Dissertation

Sentiment Analysis in the Arabic Language Using Machine Learning

Submitted by

Saud Saleh Alotaibi

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2015

Doctoral Committee:

Advisor: Charles W. Anderson

Asa Ben-Hur
Indrakshi Ray
Chris Peterson

Copyright by Saud Saleh Alotaibi 2015

All Rights Reserved

Abstract

Sentiment Analysis in the Arabic Language Using Machine Learning

Sentiment analysis has recently become one of the growing areas of research related to natural language processing and machine learning. Much opinion and sentiment about specific topics are available online, which allows several parties such as customers, companies and even governments, to explore these opinions. The first task is to classify the text in terms of whether or not it expresses opinion or factual information. Polarity classification is the second task, which distinguishes between polarities (positive, negative or neutral) that sentences may carry. The analysis of natural language text for the identification of subjectivity and sentiment has been well studied in terms of the English language. Conversely, the work that has been carried out in terms of Arabic remains in its infancy; thus, more cooperation is required between research communities in order for them to offer a mature sentiment analysis system for Arabic. There are recognized challenges in this field; some of which are inherited from the nature of the Arabic language itself, while others are derived from the scarcity of tools and sources.

This dissertation provides the rationale behind the current work and proposed methods to enhance the performance of sentiment analysis in the Arabic language. The first step is to increase the resources that help in the analysis process; the most important part of this task is to have annotated sentiment corpora. Several free corpora are available for the English language, but these resources are still limited in other languages, such as Arabic. This dissertation describes the work undertaken by the author to enrich sentiment analysis in Arabic by building a new Arabic Sentiment Corpus. The data is labeled not only with

two polarities (positive and negative), but the neutral sentiment is also used during the annotation process.

The second step includes the proposal of features that may capture sentiment orientation in the Arabic language, as well as using different machine learning classifiers that may be able to work better and capture the non-linearity with a richly morphological and highly inflectional language, such as Arabic. Different types of features are proposed. These proposed features try to capture different aspects and characteristics of Arabic. Morphological, Semantic, Stylistic features are proposed and investigated. In regard with the classifier, the performance of using linear and nonlinear machine learning approaches was compared. The results are promising for the continued use of nonlinear ML classifiers for this task. Learning knowledge from a particular dataset domain and applying it to a different domain is one useful method in the case of limited resources, such as with the Arabic language. This dissertation shows and discussed the possibility of applying cross-domain in the field of Arabic sentiment analysis. It also indicates the feasibility of using different mechanisms of the cross-domain method.

Other work in this dissertation includes the exploration of the effect of negation in Arabic subjectivity and polarity classification. The negation word lists were devised to help in this and other natural language processing tasks. These words include both types of Arabic, Modern Standard and some of Dialects. Two methods of dealing with the negation in sentiment analysis in Arabic were proposed. The first method is based on a static approach that assumes that each sentence containing negation words is considered a negated sentence. When determining the effect of negation, different techniques were proposed, using different word window sizes, or using base phrase chunk. The second approach depends on a dynamic

method that needs an annotated negation dataset in order to build a model that can determine whether or not the sentence is negated by the negation words and to establish the effect of the negation on the sentence. The results achieved by adding negation to Arabic sentiment analysis were promising and indicate that the negation has an effect on this task. Finally, the experiments and evaluations that were conducted in this dissertation encourage the researchers to continue in this direction of research.

Acknowledgements

In the name o f Allah, the Most Gracious and the Most Merciful

I thank Allah for granting me the patience, health, guidance and determination to complete this dissertation successfully. This dissertation owes its existence to the help, support, and inspiration to many people. Firstly, I acknowledge both Colorado State University and my country, Saudi Arabia, for supporting me and giving me the opportunity.

I would like to express my sincere appreciation and gratitude to Dr. Charles Anderson for his support and encouragement during my work for this dissertation. He directed me to a broad range of resources on the web and in his library. He answered all of my questions and helped me to narrow my search. I thank him for his constructive and positive criticism and for patiently reviewing my work. I also wish to thank the other members of my dissertation committee, Dr. Asa Ben-Hur, Dr. Indrakshi Ray and Dr. Chris Peterson, for their guidance, careful reading and comments. I also thank and remember Dr. Robert France, whom one of the committee and he passed away before ending this work.

I owe special gratitude to my family. Words cannot express how grateful I am to my father who passed away in January 2014, my mother, and my sister Mona who died on December 2012, brothers, and sisters in my home country. A special thanks to my family who live with me in United States, my wife, sons and my lovely daughter for their continuous and unconditional support of all my undertakings, scholastic and otherwise. Lastly, for all my friends who support me at Colorado State University and city of Fort Collins, I give them my thanks.

This dissertation is typset in LaTeX using a document class designed by Leif Anderson.

Table of Contents

CHAPTER 1

Introduction

With the growth of the internet as a means of communication between people, many modern methods have been established in order to allow people to indulge themselves more in this form of communication. As a result of this phenomenon, increasing numbers of opinions and thoughts are being spread and published over the internet. According to Pew Internet & American Life Project Tracking surveys (2014), around 87% of American adults use the internet. In addition, 87% of online adults claimed that the internet helps them find and learn new information (Purcell, 2014). Around 81% of them use it to browse information about products and services that they are thinking of buying (Purcell, 2014).

Another study also showed that 73% of online adults use social networks such as Facebook, LinkedIn and Google Plus (Duggan and Smith, 2013). From Twitter and Facebook to online shopping and Forums, website contains numerous opinions, thoughts and sentiments. Facebook has 1 billion active monthly users who share around 3.5 billion items of information (posting text, images, etc.) (Zuckerberg, 2012). Twitter has more than 140 million users who generate around 340 million tweets daily (Twitter, 2012). In addition, user reviews, which are found on many market websites, may be considered a good source which help to build people's opinion about specific topics.

The internet contains a wealth of information that people can use to help them make a decision about a given issue. People usually try to ascertain other people's opinions that are found online about products, countries that they are considering traveling to and spending time in, or movies that they are thinking of watching in a cinema. As a result of that, much opinion and sentiment about specific topics could be collected and analyzed from

these websites. Therefore, the need to automate the process of text sentiment analysis has now arisen. It will be helpful for people to be able to access opinions and sentiments about a specific topic in a reasonable manner, rather than making them search for and read reviews in order to obtain a final opinion. For example, if somebody wants to buy a specific type of digital camera, such as a Canon, and is still not sure about it, he or she can surf the internet and read customer reviews about the product. Eventually, a decision can be made depending on these reviews. This manual process is a kind of opinion mining or sentiment analysis.

Sentiment analysis has proved beneficial for several natural language processing (NLP) tasks such as answering systems and information extraction (Pang and Lee, 2008). The information extraction (IE) aims to extract a piece of information that is relevant to a specific topic or user's needs. For example, people tend to use the internet nowadays to broadcast their thoughts and ideas about topics or issues by using forums or other social networks. Some of these ideas are positive, while others are more violent in manner and content.

This notion of spreading sentiment online has created a new area in text analysis, expanding the subject of study from traditionally fact- and information-centric views of text in order to enable sentiment-aware applications. Over the past decade, the extraction of sentiment from text has attracted a lot of attention, both in industry and academia. Formally, sentiment analysis attempts to establish people's opinion from their writing. Many fields are included in this topic, such as natural language processing, machine learning, and computational linguistics.

The Arabic language is one of the most widely used languages, spoken and written by more than 220 million people in over 57 countries (Lewis, 2009). It is not like European languages, such as English, because of its richer morphological structure. It also has many challenges that require special treatment. Therefore, Arabic natural language processing has become attractive to researchers due to its complexity and the scarcity of available resources; as a result, the importance of addressing this language has been noted. It can be seen that strong effort is being made with the fundamental tools of NLP in Arabic, such as the morphological analyzer, part of speech tagger, and syntactical parser. According to Farghaly and Shaalan (2009), the field of Arabic NLP is still at an early stage of evolution. Nevertheless, work in some areas, such as sentiment analysis, is beginning to appear.

Choosing to work with the Arabic language is due to several factors. Firstly, Arabic sentiment analysis has been needed due to its large scale audience who use the online resource nowadays. Secondly, interesting and challenging points behind this language relate to its history, strategic importance to its nation, and its culture and heritage. In addition, the limitation that the language has in this field starts from the resource and ends to the tools.

1.1. Objective of Dissertation

This section explains the main motivation behind this dissertation. The first part details the research questions that this work aims to address, and the second presents the hypotheses that may be raised by the research questions.

This work will aim to address the following major questions.

Research Question 1: Are there enough sentiment corpora for the Arabic language?

- Is there a need for more free annotated data for Arabic sentiment analysis?
- What are the domains and language types of the available Arabic sentiment corpora?

Research Question 2: How should a highly inflectional and morphological language such as Arabic be treated in sentiment analysis?

- What features work best in Arabic in terms of document or sentence sentiment analysis?
- Does the Arabic language need a nonlinear classifier algorithm other such as Neural Networks (NNs) than the one commonly used which is the Support Vector Machine (SVM), due to its complexity?
- Are there differences between dealing with Modern Stander Arabic (MSA) and Dialect Arabic (DA) in the case of choosing features and the machine-learning algorithm (the classifier)?
- Could applying cross-domain mechanisms improve the process of the Arabic sentiment classification because of resource limitation?

Research Question 3: What is the effect of negation in Arabic sentiment analysis?

- What is the best method of accounting for negation with Arabic sentiment analysis?
- Does negation differ between Dialect Arabic and Modern Standard Arabic?

The motivation behind the first question is the investigation of the resource (sentiment corpus) availability in this field (Arabic sentiment analysis). Sentiment analysis is relatively new in Arabic compared to other languages, such as English. If there is no free public resource in this domain, the research in this field will be unable to develop quickly. In addition, sentiment classification is a very domain specific problem (Aue and Gamon, 2005). Therefore, the more domains of annotated sentiment corpora there are, the greater understanding of the sentiment can be obtained. This corpus should also include different types of Arabic

language that include Modern Standard as well as Dialect. This would help to show the different styles and words that express sentiment orientation in a better manner.

The second research question investigates which machine-learning algorithm may best analyze the sentiment with Arabic. Do the same methods that are used in English also work well with Arabic, or does Arabic require other methods and machine learning algorithms to deal with the complex nature of the Arabic language? In the case of both classification levels: document and sentence, the best features that work in each level must be found. Dialect Arabic (DA) needs specific treatment, as most of the basic NLP tools, such as part-of-the-speech tagger, only work with MSA. Using different external resources to cooperate with the traditional feature model may improve the accuracy of the classification. Lastly, the needs of using the existence annotated data in limited resource language and applying this to a new domain is another method to save time and effort of annotation process.

The role of negation in Arabic sentiment analysis is expressed in the third research question. Many other works study the effect of negation in detail in the English language, while few Arabic studies touch this issue as this field is still at an early stage. How does negation work in either Modern or Dialect Arabic, and how does it cooperate with sentiment? Finally, what is the best method to inject the negation while analyzing Arabic sentiment text using machine learning Classifier?

The main hypotheses that are claimed by this dissertation are presented below.

- *Hypothesis 1*: Not enough free corpora are provided to the research community for sentiment analysis in the Arabic language.

- *Hypothesis 2*: The Arabic language needs a more variety of features and representation, such as syntactic, semantic and stylistic features, in order to capture the sentiment orientation.
- *Hypothesis 3*: Very recently developed methods in Natural Language Processing (NLP) application, such as word clustering and SentiWordNet, may be helpful for sentiment analysis in Arabic. The word clustering in the Arabic language helped in other NLP applications such as name entity recognition. Therefore, it could be a helpful feature in improving the performance of the machine learning algorithm in sentiment analysis for the Arabic language.
- *Hypothesis 4*: Different machine learning algorithms such as Neural Networks work best with a highly inflectional and morphological language such as Arabic.
- *Hypothesis 5*: Applying cross-domain in the field of Arabic sentiment classification would have a big impact on the performance of the classifier and save the time and effect of labeling a new domain with sentimental tags.
- *Hypothesis 6*: Having an awareness of the negation while analyzing the sentiment in the Arabic language leads to the best performance.

1.2. Contributions

The approach taken in this dissertation to address the above questions and investigate the hypotheses results in the following contributions.

In order fill the limitation of the resource in Arabic sentiment analysis, we start this work with building our annotated corpus for Arabic sentiment data. Our corpus contains different types of Arabic language, Modern Standard, and Dialect. We made the annotation at two

different levels and types that not have been done before in Arabic sentiment analysis field. This corpus then is used during our work. This is explained in details in Chapter 3

Multiple features are proposed and investigated. Our work with this part tries to find the most suitable features that might work better with Arabic in different types (subjectivity, or polarity) and levels (document or sentence) of classification. Some of these features are proven to be worked better in a particular level of classification, such as document levels. In addition, the other of features shows their benefits while they are used in particular classification types, such as subjectivity classification. For example, the performance of the classifier was improved by 4% in the case of subjectivity document level classification. Moreover, two new features, which are the polarity score and the word clustering ID, are proposed and used with Arabic sentiment analysis. There features add a more semantic aspect of the words to the features model that helps the sentiment classification in Arabic. Most of these proposed features are examined with different classifiers including Bayesian, linear and nonlinear types. This work considered the first one that includes these varies with features and classifier in Arabic sentiment analysis that helps the new researchers in this field and provides a baseline to this area. This contribution is described in Chapter 4 and 6.

This dissertation also gives a new proposed method to be considered with Arabic sentiment analysis. The negation plays a main role in the sentence by flipping the meaning of the words. We provide a comprehensive solution to this issue by generating the negation words list and proposing new methods to how injecting the negation effects with Arabic sentiment analysis. The details of proposing negation method is explained in Chapter 5.

In Arabic sentiment analysis, there is a limitation in the resource. This dissertation provides the shows the availability of using the cross-domain technique with the sentiment

classification process with Arabic. By applying this method, we do not need to consume more time and effort to annotate a new type of data domain. The experiment results show the promising of using this technique with Arabic sentiment analysis. The work related to this approach is described in Chapter 8.

1.3. Dissertation Structure

In this chapter, we have presented the research problem of Arabic sentiment analysis. The main objectives of this dissertation are also explained. The remainder of the dissertation is organized as follows.

Chapter 2 summarizes the background of both sentiment analysis and the Arabic Language. The last part of this chapter discusses the related works in Arabic sentiment analysis.

Chapter 3 introduces the new Sentiment corpus that we built in Arabic. The details of building this corpus is explained followed by showing the characteristics of our corpus. The last section in this chapter explains the Arabic Morphology tool that is used in preprocessing this corpus.

Chapter 4 explains the details of analyzing Arabic sentiment in texts. It introduces the proposed features as well as using different machine learning Classifiers. This chapter gives a comprehensive investigation about using different features and classifiers within different types (Subjectivity, Polarity) and levels (Document, Level) of Arabic sentiment analysis.

The concept of negation and its effect on the Arabic Sentiment field is introduced in Chapter 5. This chapter shows the negation concept in the Arabic language and its importance to the sentiment field. It also explains the details of the proposed methods that tackle the problem of negation in Arabic sentiment analysis.

Chapter 6 introduces examining the Neural Networks (NNs) classifier for Arabic Sentiment. This chapter shows the methodology of using NNs and the results of experimenting with them.

Chapter 7 discusses the different mechanisms used to learn sentimental knowledge from one domain and apply it to a different one. This concept is called the cross-domain process, which helps in the case of the limited resources.

Chapter 8 concludes the final remarks of the dissertation. This chapter ends with suggested directions for future improvements and research.

1.4. Chapter Summary

This chapter gives introductory information about the topic of this dissertation. It starts with explaining that a lot of information appears online and how individuals, businesses and governments could use and benefit from this online knowledge. One of these benefits is obtaining the people's sentiment from their writing. Many fields are involved in the process of sentiment analysis including natural language processing, machine learning and computational linguistics. Human languages have different features and characteristics that are different from one language to another. For this reason, sentiment techniques that work well with one language may or may not work with another one. The most investigated language in sentiment analysis is the English language. There is a small amount of work done in the Arabic language. However, there is a reasonable portion of online knowledge as well as Arabic speakers over all the world. According to the Internet World State (2013), the Arabic language is considered to be among the top ten languages uses in the Internet. Therefore, more work is needed in the field of sentiment analysis, especially for Dialect Arabic. This chapter ends with the motivation, research questions, and hypothesis that

this dissertation tries to address. The next chapter gives background information about sentiment analysis techniques as well as explains essential terms in Arabic as a language and as NLP basis terminologies.

CHAPTER 2

Background and Related Works

This chapter explains the basic information relating to our work. The first section shows the definition of Sentiment as well as the different techniques as they have been used for English sentiment analysis. The main concepts about the Arabic Language are illustrated in the second part of this chapter. The related works that have been achieved in the field of sentiment analysis for Arabic is mentioned in the last section.

2.1. Sentiment Analysis

Before sentiment analysis is discussed with regards to any language, the definition of "sentiment" first needs to be agreed on. Many terms are found in the literature that are used interchangeably to refer to this concept including opinion, subjectivity in a text, sentiment, emotion, evaluation, belief and speculation. All these terms refer to a private state which is not open to objective observation or verification (Quirk et al., 1985). This diversity of the terms also has the effect of the computation analysis areas being known as opinion mining, sentiment analysis or subjectivity analysis (Pang and Lee, 2008). This sometimes creates ambiguity for the reader or beginners in this field.

Generally, the textual information falls into two categories. The first category is the factual information that only contains facts, objective expression about entities, or events. The second category is the subjective information that shows the actual feeling, opinion of the writer toward entities and events. In this dissertation, view the concept of sentiment as the subjective information that shows the feelings or opinions of a person about a specific topic or subject (Turney, 2002).

Sentiment analysis is a method of capturing the sentiment (feeling or opinion) of people towards a specific topic. This field may be considered part the machine learning, natural language processing and computational linguistics. In other words, it usually tries to evaluate and extract the sentiment of people from their writing. In literature, SA has many names, including subjectivity analysis, opinion mining, review mining and appraisal extraction (Pang and Lee, 2008). Moreover, the sentiment of the text can be explicit or implicit. If explicit, a text directly gives a sentiment, such as (**إنها سيارة رائعة** / *InhA syArħ rAŷςħ* / It is a nice car)[1], while if implicit; the text implies a kind of sentiment like, (**عمل الشاحن لمدة اسبوع فقط** / *ςml AlšAHn lmdħ Asbwς fqT* / The charger only works for one week). More formally, SA can be defined as:

Given a text t from a text set T, computationally assigning polarity labels p from a set of polarities P in such a way that p would reflect the actual polarity that is found in T (Pang and Lee, 2008).

In sentiment analysis, the first step aims to determine or classify whether the content of the text is subjective or objective (Pang and Lee, 2008). This task is called subjectivity classification. The second task is the analysis of the subjective text in order to determine which of the sentiment polarities it has (Pang and Lee, 2008). The strength of this polarity varies from one opinion to another. One example of this is that user reviews about some product need to be categorized as positive or negative toward the target. This shows binary polarity. The work will be more difficult when the polarity is expanded to include more than two items, such as if the neutral class is added. Another type of sentiment includes emotions

[1]Throughout this work, Arabic sentence is represented in three variants: (Arabic sentence / transliteration scheme (Habash et al., 2007) / English translation)

such as Sorry, Hugs, You Rock, Wow, etc. (Socher et al., 2011). Here, the task becomes a multiple class classification challenges.

The classification process can be carried out at different levels of the text: term, phrase, sentence or document. The output results from each level are usually used as input for the next level. For example, the output of sentence evaluation is used and expanded for the document classification. Our work would be concentrated to find the sentiment at the sentence and document level.

Another type of sentiment analysis is one that deals with the sentiment target or the discovery of the sentiment target. Most work that has been done in the sentiment analysis field relates to finding sentiments regarding a general topic or target, such as user reviews on a movie or product. In such reviews, it is easy to determine the topic, as there is an assumption that the review talks about a specific product. Conversely, it is more difficult in the case of an unknown target, such as with feature-based sentiment analysis. It is difficult to establish what features of the product the user has written about, and then to determine the user's opinion of it. Therefore, an exploration is first made to establish what features that a user has written about by using feature extraction approaches (Popescu and Etzioni, 2005). The next step is to determine the sentiment or opinion of these features. This kind of process in sentiment analysis is not considered in this dissertation.

Sentiment classification is applied in different domains. The most famous domains are movie reviews and customer reviews in a market domain. Much research has been done on these areas (Pang et al., 2002; Turney, 2002). News is another domain that has been investigated by researchers (Abdul-Mageed et al., 2011; Bautin et al., 2008). The type of data that is used in sentiment classification differs from one domain to another, as well as

from language to language. In other words, a sentiment analysis system that works well for movie reviews may not work as well with customer reviews. This issue comes from the diversity of the sentiment from one domain to another. Therefore, sentiment classification is a very domain-specific problem (Aue and Gamon, 2005).

2.2. Challenges of Sentiment Analysis

Generally, sentiment analysis or classification is considered a special case of text classification in a natural language processing. Although the number of classes in sentiment analysis are small, the process of sentiment classification is more difficult than the traditional Topic Text Classification (Pang and Lee, 2008). In Topic Text Classification, classification relies on using keywords, but this dose not generally work well in the case of sentiment analysis (Turney, 2002).

The other difficulties in sentiment analysis come from the nature of this problem. Sometimes, the negative sentiment might be expressed in a sentence without using any obvious negative words. Moreover, there is a fine line between whether a sentence should be labeled objective or subjective. Determining the opinion holder -the one who expresses the sentiment in the text- is one of the most difficult tasks in sentiment analysis. The sentiment analysis highly depends on the domain of the data. The words sometimes have positive sentiment in a specific domain, whereas they have another polarity sentiment in a different domain (Pang and Lee, 2008). Finally, some other writing styles such as irony, sarcasm, or negated sentences could bring more challenges to sentiment analysis.

2.3. Application of Sentiment Analysis

In a marketplace, businesses realize the importance of the internet in gathering users' opinions and reviews about their products and services. Time is more valuable to businesses than to normal users. Normal users often spend some time surfing the internet in order to establish the opinions of other users, while businesses generally need an automated system that can help them ascertain the sentiments and opinions of users of their products and services. A tool that can obtain and analyze user reviews in order to understand the final sentiment is more valuable to businesses. This tool may provide them with the feelings of customers and ideas that help them to improve their products and services.

The World Wide Web provides a great place that the people gain knowledge from the information. There is no need to ask a friend when you are wanting to buy a product, going on a vacation, or needing some services. The only thing that you need is the internet to surf through this unstructured information. Therefore, sentiment analysis should be able to surf this information and bring it in structured format to the end users.

Nowadays, people tend to use the internet to broadcast their thoughts and ideas about topics or issues by using forums or other social networks. Some of these ideas are positive, while others are more violent in manner and content. According to Glaser, et al.,(2002), extremist groups use the Internet to spread hate and violence among other groups. Therefore, sentiment analysis has the potential to be more valuable in these cases in monitoring the sentiment of groups over the internet. This helps the government to discover any violence at an early stage and to begin to deal with it before it expands. Abbasi, et al.,(2008) provided a novel approach that discovered the sentiment of violence in two groups: US supremacists and Middle Eastern extremist groups.

2.4. Methodologies Used for English Sentiment Analysis

A large range of approaches and techniques are used to investigate the problem of sentiment analysis. Most of these approaches are built to deal with the English language as it is the dominant language of science. However, this should not stop researchers from building techniques that work with other languages, such as Chinese, Korean, Japanese and Arabic. This section describes the concepts and research that are used for sentiment analysis in English.

There are two main approaches that are found in the literature to analyzing sentiment. The first is a Machine Learning (ML) approach. In this method, annotated data is converted into feature vectors and used to train ML classifiers to infer a combination of specific features yielding a specific class (Pang and Lee, 2008). After this process, a model has been created and is used to predict the class of new, unseen data. The second method is a semantic approach, which is based on calculating and extracting the polarities of all sentiment words by using a Sentiment Lexicon (Turney, 2002). This lexicon contains the semantic intensity of words by indicating some value in each class. In this work, the ML approach will be followed.

This approach usually starts with a set of training data. The data should be chosen and categorized properly in order to achieve good prediction results. If not, the data requires a manual effort from the annotator to annotate the data with its subjectivity and polarity. Sometimes, the websites that contain user reviews have ratings along with the reviews. A set of data like this is called a corpus. Next, features are chosen to represent the text (the review). The next step is to train a classifier that has been chosen from the corpus, and the performance of the classifier is then evaluated on the testing data. This process is usually

repeated in an iterative manner if the initial performance is weak. During this repetition, some of the features may be fine-tuned. Some of the preprocessing carried out includes word stemming and the removal of stop words.

Three different ML approaches were investigated by Pang, et al., (2002). They employed Naive Bayes (NB), Support Vector Machine (SVM) and Maximum Entropy Classification, and inferred that the machine learning algorithms do better than human baselines for sentiment classification. In addition, the results show that the performance of the SVM was better than other classifiers. The SVM is used with specific features, including uni-gram and lemmatized uni-gram (Mullen and Collier, 2004). They showed that their approach outperformed other approaches that did not use computations for these features. A combination of classifiers was used by Prabowo and Thelwall (2009). The basic idea of this research was to build hybrid classifiers. In their work, the document that is not classified by one classifier is sent to the next classifier until either the document is classified or there are no more classifiers. General Inquirer-Based Classifiers (GIBC), Rule-Based Classifiers (RBC), Statistics-Based Classifiers (SBC) and the SVM are used in this method. It was discovered that the SVM and SBC improved the performance of the method. Finally, when comparing supervised and unsupervised approaches, Chaovalit and Zhou (2005) showed that supervised methods achieved 84% accuracy for three-fold cross validations and 77% accuracy using unsupervised methods with movie reviews.

2.5. Common Features in Sentiment Analysis

Sentiment analysis is considered a classification problem that can be solved by using the machine learning concept. Machine learning provides many algorithms that work for classification, but the challenge of finding a sentiment in a text is determining the best

features to be used. The following sections reveal the common features that are used in sentiment analysis.

Term Frequency is the measurement of how many times a specific term is repeated in a document. This has long been emphasized in traditional information retrieval systems. The term presence shows the existence of the term in the document in a binary mode. The document model here shows that term presence is 1 if the term appears at least once in a document, and 0 if not. The term presence model is used in (Pang et al., 2002) and shows improvement compared with the term frequency model.

In sentiment analysis, it is important to find the adjectives, as these are good indicators conveying the sentiment orientation in the text (Benamara et al., 2007). Using the part-of-speech (POS) tagging system decreases the ambiguity of the word (Wilson et al., 2009). When a word is annotated with its POS tag, this helps to increase the NLP system's confidence in its actual meaning. This will help significantly in the case of more morphological languages such as Arabic. For example, the word (**جمل** / *jamal* or *jam ala*) could be the noun "camel" or the verb "make something beautiful". The POS will help to determine the correct meaning of the word. Turney (2002) used the POS feature for adjectives and adverbs in order to obtain the sentiment orientation at document level.

Some other features, such as the style of the text, may contribute to the sentiment orientation of the text. The stylistic features include the length of the sentences, the length of the words, special characters, richness of words, etc. Some research has shown the effect of using the length of sentences as a feature in sentiment analysis (Na et al., 2004). In addition, Abbasi, et al.,(2008) investigated more than one stylistic feature in multi-language sentiment analysis, including English and Arabic. They found, for example, that in both

Table 2.1: An example of Arabic sentence

Arabic Text	تأسست جامعة ولاية كولورادو الحكومية سنة ١٨٧٠ ككلية كولورادو للزراعة
Transliteration	*tOsst jAmçħ wlAyħ kwlwrAdw AlHkwmyħ snħ* 1870 *kklyħ kwlwrAdw llzrAçħ*
Translation	Colorado State University was established in 1870 as the Agricultural College of Colorado.

languages the positive sentiment text is shorter than the negative sentiment text in terms of the total number of characters. They also found that using stylistic features in addition to other features increases the performance of sentiment analysis in web forum discourse (Abbasi et al., 2008).

2.6. Arabic Language

The Arabic language is comprised of 28 letters (25 consonants and three long vowels). It is a cursive language, in which words consist of cursive Arabic letters connected to one another. In addition, the writing in Arabic runs from right to left. Like other languages, such as Japanese and Korean, Arabic has no capitalization. Table 2.1 illustrates an example of Arabic text beside its translation and transliteration. Arabic letters have different shapes depending on their position in the word. Unlike English, which has dedicated letters to represent short vowels, Arabic has diacritics that play the same role as short vowels in English and determine the pronunciation or the sound of the letter. All words in Arabic are derived from a root which is composed of constants. These are generally three or four letters called radicals (Daya et al., 2007).

There are two types of Arabic sentences, nominal and verbal, these are determined by the part-of- speech of the first word in a sentence. A nominal sentence has no verb. It is formed of a subject and a predicate. These vary from very simple forms to more complicated

sentences. The simple nominal sentence consists only of nouns and adjectives, whereas the subject is composed of two words, and the predicate is another sentence within a complicated one (Ryding, 2005). A noun in the Arabic language may come in three numbers: singular, dual, and plural. On the other hand, verbal sentences start with a verb and follow different structures and orders. The standard structure of the verbal sentence is Verb-Subject-Object (Ryding, 2005). There are past, present, and future verb tense as well as imperative, perfect, and imperfect action in Arabic language. These basic features introduce new challenges in NLP perspective; therefore, different techniques will be needed in order to achieve comparable performance level to what has been achieved in other language such as English.

There are three main types of Arabic. These types are Classical Arabic CA, Modern Standard Arabic MSA, and Dialect Arabic DA (Farghaly and Shaalan, 2009). CA is the oldest version of Arabic, which is used in the earliest age of Arab nation. The MSA is the formal Arabic language, which is used nowadays in education, books, newspapers, media, and even as the official language of Arabic countries. DA is a kind of colloquial language that differs from region to region in Arab countries. There are similarities between MSA and CA since MSA is based on the same syntax and morphology of CA (Ryding, 2005), but there are many differences between MSA and DA. However, the DA share with MSA because most of the DA words derive from MSA. These differences between types will, therefore, affect the building of Arabic NLP tools, as the tools that are built for MSA may not work with equal efficiency for DA. Therefore, there is a need to build a native tool that works especially well with DA (Farghaly and Shaalan, 2009). In the remainder of this dissertation, the word (Arabic) will be used to refer to MSA unless otherwise specified.

Table 2.2: An example of word derivation process in Arabic language

Word Root	(كتب / ktb / 'Write')	
Pattern	(مفعول / $mfςwl$ / pattern)	(فاعل / $fAςl$ / pattern)
Generated Word	(مكتوب / $mktwb$ / 'Has been written')	(كاتب / $kAtb$ / 'writer')

2.6.1. Root and Pattern in Arabic. The key concepts in Arabic morphology are the concepts of root and pattern, which interlock to form the final shape of the word. Roots that are mainly three and four (and rarely five) radicals, i.e., consonants, comprise the smallest meaningful language unit (Daya et al., 2007; Farghaly and Shaalan, 2009; Ryding, 2005). The pattern is the group of letters that have been used to derive the words. These two features possess the lexical and grammatical meanings, respectively. Early studies of Arabic morphology (Beesley, 1996) show that Arabic has almost 5000 roots while another (Darwish, 2002) has estimated that the roots of nouns and verbs together comprise 10,000. There are also around 400 different patterns in Arabic (Beesley, 1996) that may be added to the root.

These two features show how the root of Arabic words can form a variety of word forms that are the derivation and inflection. The derivation is the process of word/lemma formation from its root (Ryding, 2005). This process occurs by combining a specific consonantal root with a desired pattern. Table 2.2 shows an example of this process. The inflection of the Arabic word is caused by the contextual position of a word (Ryding, 2005). Each different type of words (noun, verb or pronoun) have different inflectional categories that may be applied to them. In the case of nouns and adjectives, four inflectional categories are applied: gender, number, case and definiteness. Verbs have a larger number of features: aspect, person, voice, mood, gender and number. Finally, pronouns tend to possess four different features: person, gender, number, and case (Ryding, 2005).

The Arabic language is highly inflectional and derivable. Arabic has a small number of roots, but this increases its complexity. The agglutinative feature of the word structure adds considerable difficulty to the language morphology (Ryding, 2005). Arabic words may work with three types of affixes: prefixes, infixes, and suffixes. Affixes may be one letter long or a combination of multiple letters. In addition to their complex nature, the level of ambiguity of Arabic morphemes is notable. Determining whether a letter is an affix or part of the stem is not an easy task, especially when there is an absence of short vowels. These characteristics affect the NLP tools that deal with Arabic, such as the part-of-the-speech tagger, morphology analyzer, name entity recognition and syntactical parsing. Several studies have been conducted around this.

2.6.2. Challenges of Arabic Natural Language Processing. The absence of rigid and strict rules in adding punctuation in MSA text makes it very hard to identify the sentence boundaries (Shaalan, 2010). This issue is also a significant challenge in DA, as there are no rules governing it. People often write whole paragraphs without using punctuation, except for the full stop at the end. Literal conjunctions, such as (و /w/ and), are used to organize and link the sentences. This challenge has a direct impact on Arabic sentiment analysis, and particularly when selecting proper sentences from entire texts. There is also no capitalization in Arabic, which makes the determination of sentence boundaries a crucial and challenging task for NLP in the Arabic language, especially for the task of sentiment analysis.

The negation in Arabic text also plays a major role in NLP tasks, especially in sentiment analysis. Negation words can reverse the meaning of a sentence; as a result of which the sentiment orientation should be changed. For example, the following sentence (انا احب هذه

القصة / *AnA AHb hðh AlqSħ* / I like this story) attributes a positive sentiment to the story, whereas this sentence (انا لا احب هذه القصة / *AnA lA AHb hðh AlqSħ* / I do not like this story) negates the meaning as well as indicating positive sentiment by using word "like". These two sentences are very similar, the difference between them being only one word. However, not using negation will negate or reverse the sentiment orientation. For instance, in the sentence (لا عجب ان الجميع يحب هذه القصة / *lA ςjb An Aljmyς yHb hðh AlqSħ* / No wonder everyone loves this story), the (لا / *lA*/ No) word here does not negate the meaning. Arabic has different words that are used in negation. Some of them could be used to express another style in Arabic rather than a negation. Not taking negation into consideration will partially decreases the performance of the classifier used in sentiment analysis.

The most important thing in NLP is to have a sizable corpus of single or multiple domains. In literature, there are several standard corpora for Arabic. Unfortunately, only a few of these are open-source. Researchers with no means of accessing the standard corpora develop their data in-house (Saad and Ashour, 2010). A specialized corpus for specific NLP tasks is needed in the case of the Arabic language. For example, an annotated corpus with the subjectivity or polarity of the Arabic language needs to be developed for the task of sentiment analysis. Few attempts to do this have been made so far by researchers.

2.7. Related Works in Arabic Sentiment Analysis

Much of this research has been done in English, as this is the dominant language of science. Recently, a few researchers have concentrated on applying sentiment analysis to other languages, one such language being Arabic. Figure 2.1 shows the difference between the research that has been conducted in the Arabic and English languages. This data is

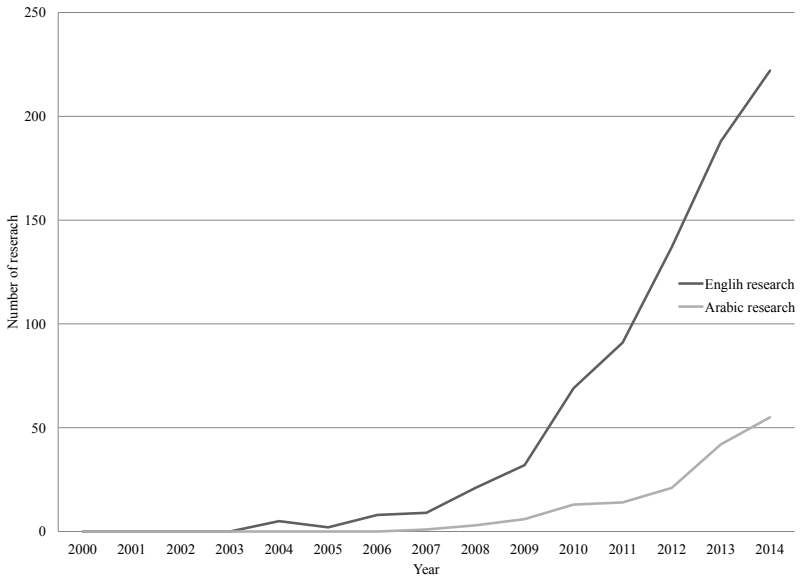


Figure 2.1: The difference in research that has been conducted in Arabic and English

collected by using relevant keywords in sentiment analysis field in both languages. The Google Scholar website is used to collect the numbers of research. For a particular keyword, the Google Scholar is used for a specific period. The results that are retrieved are shown in the top page of the Google website result. These results are used in our comparing.

It is clear that there is a big gap between the work that has been achieved in Arabic and English, Figure 2.1. This might be due to limitations in the tools or resources of the NLP of Arabic. In addition, it may reveal that Arabic requires special treatment due to its complex nature and structure.

This section summarizes related work that has been done in Arabic sentiment analysis. The summarization are organized into subsections titled to Arabic sentiment corpora, features and methods, and negation.

2.7.1. Arabic Sentiment Corpora. Arabic sentiment corpora are still in their early stages. Figure 2.2 illustrates the top ten languages on the internet. These statistics were

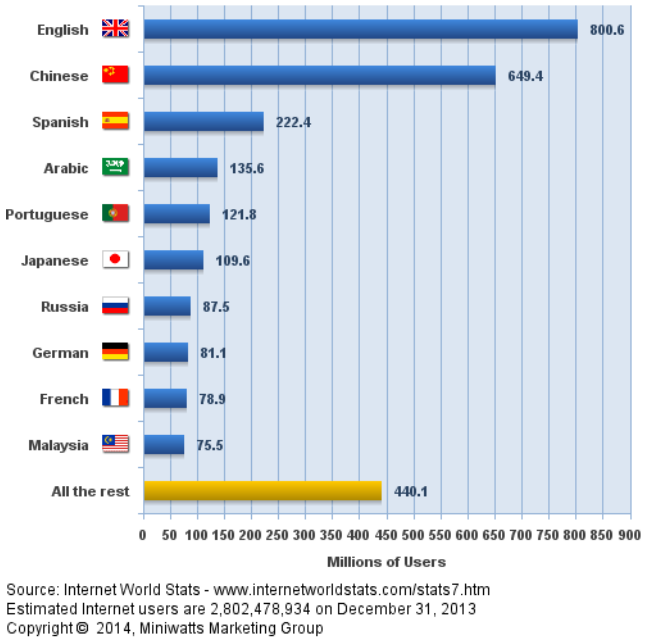


Figure 2.2: The top ten languages in the internet

captured in 2013 according to the Internet World State (2013). This may reveal why most research is conducted in English as well as Chinese; there are plenty of sources in these languages on the internet. However, the Arabic language is considered to be among the top ten languages (fourth position). A small number of research studies have been carried out in this direction. Most researchers in Arabic sentiment analysis built corpus, manually annotating it at either the document or sentence level.

The Opinion Corpus for Arabic (OCA) (which is the only published corpus) contains 500 movie reviews. They are annotated at the document level. Half the reviews are considered positive and the rest are negative (Rushdi-Saleh et al., 2011). Further work undertaken to build a multi-genre subjectivity and sentiment corpus for modern standard Arabic is called AWATIF (Abdul-Mageed and Diab, 2012a). The domain of this data was taken from a news wire in different domains (400 documents), Wikipedia talk pages (around 5342 sentences), and web forums (around 2532 threads from seven web forums). The annotation

was at the sentence level and three different conditions were used to annotate the data: (1) Gold Human with Simple Guidelines (GH-SIMP); (2) Gold Human linguistically-motivated and Genre-nuanced (GHLG); (3) Amazon Mechanical Turk with Simple Guidelines (AMT-SIMP) (Abdul-Mageed and Diab, 2012a). In addition, the authors attempted to build a labeled social media corpus for subjectivity and sentiment in the Arabic language in the SAMAR project (Abdul-Mageed et al., 2012). The data was collected from four different types of social media. These included Arabic chatting, tweets, Wikipedia Talk, and forums. This corpus was a mix of long and short sentences, as well as MSA and some of DA. They provided stand-off annotations on top of the Arabic Tree Bank ATB[2] part 1 version 3 which is only free for the user who subscribes with the LDC[3] since 2003.

2.7.2. Features and Methods. Abbasi, et al.,(2008) proposed a system for sentiment analysis task in a multi-language web forum at document level. The system depends on an Entropy-Weighted Genetic Algorithm (EWGA) to choose the best features, and the SVM with linear kernel for the sentiment classification. Their method tries to find an overlap between language-independent features, including syntactic and stylistic features. The syntactic features include POS only for the English language, not for Arabic. In order to evaluate the performance of their method, the authors measured the accuracy of the classifier by dividing the number of correctly classified documents by the total number of documents. In this case, a more accurate measurement was required to help evaluate the method in both classes. The authors reported that syntactic features achieved a higher result than the stylistic ones. When the two features were employed together using EWGA, the accuracy result increased to 93.6% in the Middle Eastern forum domain.

[2]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2005T02
[3]http://www.ldc.upenn.edu/

The work of Rushdi-Saleh, et al.,(2011) focused on investigating two ML classifiers, Naive Bayes and Support Vector Machine, with two different weighting schemes (term frequency and term frequency-inverse document frequency) and three n-gram models. The effect of using the stem of the Arabic work was also investigated with different n-gram models. The authors built their sentiment corpus by collecting around 500 Arabic movie reviews from different websites. They reported an accuracy of 90.6% using the SVM with the tri-gram model and with no stemming for document level classification. In addition, they claimed that there was no big impact of using TF or TF-ID as a weighting scheme, which makes sense because both schemes represent the count of the term over the document. It could be useful to compare the presence of the term versus the term-frequency scheme.

El-Halees (2011) proposed a combined classification approach for document level polarity classification in Arabic. His method applied three different classifiers in a sequential manner: a lexicon-based classifier, a maximum entropy classifier and the K-Nearest Neighbor classifier. The result from one classifier was used as training data for the next. The text was manipulated before using the first classifier by removing the stop words. Some Arabic letters were normalized and some misspelled words corrected. A simple stemmer was used here to generate the stem of the Arabic words and TF-IDF was used as the term-weighting scheme. The F-measure was used as the evaluation metric. The F-measure that was reported in this method was between 75% and 84%, depending on the domain of the data. The average of the F-measure was also calculated; this was 82% for the positive document and 78% for the negative one. The main issue for this study was that there were no more features added to the classifier that could help to increase the performance and accuracy.

Other studies have attempted to investigate the linguistic features of Arabic and to combine these with an ML classifier in order to perform sentiment analysis. One such study tried to analyze the grammatical structure of Arabic (Farra et al., 2010). This work attempts to analyze the sentiment at the sentence level first, and then to use the results to analyze the sentiment at the document level. At the sentence level, the researchers compared two different approaches. The first was generalizing the Arabic sentence into a general structure that contains the actor and the action. The second approach used some semantic and stylistic features. The researchers used different classifiers for a different approach. They used the SVM for the grammatical classifier, and obtained an accuracy of 89%, while the J48 decision tree was used with the semantic approaches and achieved an accuracy of 80% when the semantic orientation of the words extracted and assigned manually were used, and 62% when the dictionary was used.

Another work, which investigated the effect of language-independent and Arabic-specific features on the performance of the classifier, was conducted by Abdul-Mageed, et al.,(2011). They performed two kinds of sentence level sentiment analysis for two different domains: news and social media domains. The SVM was used to classify both the subjectivity and polarity of the sentences with different features, including N-gram, adjective features and a unique feature, where all words occurring fewer than four times were replaced by the token "UNIQUE", and MSA morphological features (person, gender and number). By using different stemming and lemmatization settings with different types of independent language and Modern Standard Arabic morphology features, the researchers achieved an F1 result of 72% for subjectivity and 96% for the polarity with stem, morphology setting and ADJ features using the newswire domain. In SAMAR (Abdul-Mageed et al., 2012), they investigated the

effect that the standard features and the genre-specific features had on the subjectivity and sentiment classification of the Arabic social media domain.

2.7.3. Negation in Arabic Sentiment Analysis. Little work has been undertaken in Arabic in order to address the issue of negation, either in the negation detection problem itself or the effect of negation in sentiment analysis.

Elhawary and Elfeky (2010) considered the negation concept in their work. They relied on the Arabic lexicon to calculate the sentiment orientation score of each word or phrase. While the counting process is running, the negated word of the phrase is flipped. There are two main issues here in this work. Firstly, the authors did not mention the Arabic negation words used, stating only that they used around twenty words as negation words. Secondly, there is the issue of how they determined the negated words or phrase that come with the negation word in the sentence. This might affect the process of sentiment analysis, since it has the possibility of changing the polarity (i.e. its polarity type and strength). A further limitation of this work is that the sentiment orientation was calculated depending on the Arabic lexicon, rather than using machine learning to classify the sentiment.

Farra et, al.,(2010) also considered negation while attempting to capture the sentiment of Arabic text. The negation issue is considered in this work by only counting the frequency of the negation words in the sentence while attempting to build a semantic feature of the sentence depending on Arabic sentiment lexicon. The used features were the frequency of each positive, negative, neutral word, special character and the frequency of the negation words. The authors do not consider the ways in which words might be affected by the negation words. This resulted in a lower accuracy when compared to other methods used by the authors. As in the previous work, the authors here did not mention the list of negation

words used. In addition, relying on a simple representation (i.e., frequency counts of negation words or polarity words) would not capture all the semantics and syntax of the sentence that might be useful in sentiment classification.

Hamouda and El-Taher (2013) attempted to build a sentiment analyzer for comments on Arabic Facebook news pages. They compared different machine learning algorithms with different features. One of these was dealing with negation in Arabic. They counted only five different negation words, whereas there are many more than these, even without counting negation words in the dialects. They only added the percentage of negation words in either the post or the comment as the feature, without considering the effect of negation on the word or phrase. They claimed that adding negation word features besides the features of all words in the posts and comments gives the best performance. The general issue here is that their proposed method may work only for the domain that they have chosen, which is the posts and the comments in Arabic Facebook news pages. This might, or might not, work with regular Arabic sentiment analysis.

2.8. Challenges and Gaps in Arabic Sentiment Analysis

The studies that have been done in Arabic sentiment analysis pose some of issues. This section describes the gaps and issues in the previous studies starting with the corpus and ending with the negation concept.

It is obvious that only a few studies have been carried out on the use of Arabic corpora for sentiment analysis. Furthermore, the work that has been done has a number of limitations. First, the type of Arabic used in these corpora is MSA. The SAMAR project begins to address DA and is considered to be the first work to do so. The second issue is that all these corpora concentrate on one type of sentiment: binary polarity. No work has been

undertaken on fine-grained polarity or emotional sentiment. Moreover, most of the corpora are on the same domain, either news or user reviews (on businesses or movies) except one work undertaken by Abdul-Mageed, et al., (2012) in the SAMAR project which contains four different web genres.

Most of the research worked with one type of Arabic language, which is Modern Standard Arabic (MSA). Only one work began highlighting and investigating Dialect Arabic (DA) (Abdul-Mageed et al., 2012). In addition, there are two problems with this research. The first is the variety of DA used. The study that considers DA only included one form of DA, such as Egyptian Dialect. Each dialect contains different words and expressions that may differ in expressing subjectivity or sentiment orientation. It cannot be ensured that a method that works with Egyptian Dialect would work well with other dialects. The problem that relates to the dialect language is the lack of resources and tools. There are not enough sentiment corpora for the different dialects available to be used. In addition, the Arabic NLP tools that deal with basic NLP tools, such as POS tagger and morphology analyzer, are not yet mature, and are sometimes non-existent for the DA. Therefore, further investigation of the DA is encouraged in subjectivity and sentiment analysis in order to establish which features and ML algorithms work well with DA.

The size and domain of the data sets that are used in subjectivity and sentiment analysis are other issues. Despite some studies reporting high accuracy, this may not always reflect perfection in the proposed method, but may instead be a result of the small size of the dataset used in the experiments. In addition, some of the studies only considered either the news wire or the movie reviews domains. However, what happens if other domains are considered, such as business reviews or even different sub-domains within the main domain,

such as different types of news? Moreover, the features or the ML algorithms that work in one domain may not work with the same efficiency in other. It may be useful to use a multi-domains in order to find generalization features and methods that may work with the same efficiency for other types of data domains.

The method used to tackle the problem of how to start classifying the Arabic language is a crucial factor. First, the preprocessing phase for Arabic in order to train the ML classifier plays the main role. Despite this, most of the studies on Arabic sentiment analysis did not explain this phase in detail. Incorrect words, letters with the same shape and effect of the word, such as "أ", "إ" and "ا", and stop words all need to be corrected, normalized or removed. This process should also be undertaken in the case of DA. Secondly, most of the proposed methods in this field used the SVM as the ML classifier with a linear kernel. The Arabic language is recognized to be a highly inflectional and richly morphological language; other classifiers may work better with this language. For example, using a nonlinear kernel with the SVM, or even using the Neural Networks, may lead to better analysis of the sentiment in Arabic, especially in the case of DA, when there is a lack of NLP recourse.

While dealing with the negation in Arabic sentiment analysis, most of the works touch the basic idea of how to add and deal with negation during the sentiment analysis processing. The negation tools and styles should be specified during the first step. Previous works either depend on basic negation form or do not mention the negation syntax that they rely on. Moreover, most research in Arabic sentiment analysis does not deal with the issue of negation words while using machine learning algorithm to solve sentiment analysis. They use to using semantic approaches by counting the number of opinioned words instead of using machine learning techniques and flip the score of negated words or counting negation

words and adding this to the total score. Therefore, this dissertation tries to come up with a comprehensive method to deal with negation by using machine learning techniques to solve Arabic sentiment classification.

2.9. Chapter Summary

This chapter reviews concepts and previous work related to this dissertation in three different areas. The first one defines the sentiment in general and explains the process of how to analyze the sentiment in a text. The process includes the features that we can get from the text and the approaches that might be used to make the classification process. The second part describe the basic information about the Arabic language as well as its features that makes this language challenges in the Natural Language Processing field. The last direction of this chapter shows the previous works in Arabic sentiment analysis. In addition, it explains some of the issues and gaps that are still open in Arabic sentiment analysis. The next chapter will explain the first step in building Arabic sentiment analysis system by creating a multi-domains sentiment corpus.

CHAPTER 3

Building an Arabic Sentiment Corpus

3.1. Introduction

There are three main aspects of the sentiment analysis field: Lexicons, Annotated Corpora, and Tools. "Lexicons" relate to words, phrases and patterns that can be used to express subjectivity. "Tools" include machine learning Classifiers that use text classification algorithms, and Natural Language Processing tools which are POS tagger, Stemmer, and Morphology Tagger. The essential part is the Corpora, which contains pieces of text annotated with their polarity. These Corpora are utilized by classification algorithms to determine the sentiment of the new text.

As explained previously in Chapter 2, there is a limitation to the available sentiment Arabic corpus. The available corpora are on one domain such as movie review, have annotation on one type of level such as on document level only, have a few samples, or even require a subscription fee in order to access the data. For this reason, a new multi-domains sentiment Arabic corpus must be built as a starting point for this work and to enrich the research community of Arabic sentiment analysis. This corpus is collected from different websites, as well as some previous work undertaken to cover different domains.

Building a sentiment corpus is not as easy as it seems at first thought. Many issues and complications must be dealt with, starting from the data collection and ending with the preparation of the data to the sentiment analysis process. This chapter explains the details of building our Arabic corpus for sentiment analysis. The first Section, 3.2, describes the domains in which the data is collected, and illustrates the process of the annotation in Section 3.3, followed by the distribution of the data in our corpus in Section 3.4. The Section

3.5 discusses how the data is prepared for experimentation by removing the unwanted data (noise) and getting the primary linguistic features of the Arabic text. The Section 3.6 shows the Arabic Morphology tools that we used to get the linguistic features about the text. The summary of this chapter is shown in the last section.

3.2. Data Preparation

The research corpus was built from five different domains, which include news, reviews of the news, user market reviews, restaurant reviews, and movie reviews. The first three domains have been taken from Arabic websites, whereas the latter two have been used in previous works by Al-Subaihin, et al., (2011); and Farra, et al., (2010). The news data has been taken from the Sabq[1] news website in five different domains (local, international, sport, economic, technical news). The reviews of the news have also been taken from the same website. The Souq[2] (considered as the Amazon market place for Arab countries) is used as a source for market reviews.

The last two data sets have been taken from previous works. The restaurant reviews have been taken from the work of (Al-Subaihin et al., 2011) which captures the review of the user concerning restaurants[3]. This dataset is annotated on two levels (document and sentence level) but considers only two polarity classes (positive and negative). The movie reviews were taken from an Arabic movie review website called filfan[4] and is used in this work (Farra et al., 2010). This data is only annotated on a document level in two classes which are the positive and the negative category. Therefore, we chose these datasets to expand our corpus. In addition, we have to redo the annotation process by annotating the

[1]http://sabq.org
[2]http://saudi.souq.com/sa-ar/
[3]http://www.qaym.com
[4]http://www.filfan.com

sentiment on both levels (document and sentence) and all three polarity classes (positive, negative and neutral).

3.3. Data Annotation

Two Arabic educated individuals have been chosen to annotate the data. Each annotator was given guidelines. First, they should determine if the document is subjective or not. Second, they had to establish the polarity of the text among three categories, these being positive, neutral, or negative. Third, the annotator should go over each sentence in the document, noting its polarity if the sentence is a subjective one. Otherwise, the sentence should be noted as objective. In general, the annotator assigns each text or sentence with four possible labels: objective, subjective positive, subjective negative, and subjective neutral.

An essential tool has been built to help annotators with this process. An essential tool has been built to help annotators with this process. We created a web-based tool that helped the annotator. We used the PHP to develop this tool. The tool brings each document and its sentences on the same page. The annotator has the ability through our tool to read and annotate each sentence for a particular document. It then allows the annotator to make the annotation on the document level. Our work may be used as annotator tool to another work to annotate another dataset.

The first step was to train the two annotators, who were then asked to work on the same dataset that contained around 15% of the sentences. During this process, the inter-agreement between them was calculated using Kappa coefficient (Carletta, 1996). The inter-annotator agreements generally showed substantial agreement in the sentiment annotation process. The result of this task is reported in Table 3.1 and is between 0.72 and 0.84. These numbers are considered an acceptable range according to reporting in previous work (Abbasi et al., 2008;

Table 3.1: Basic statistics concerning the corpus next to the inter-agreement values

	News Reviews	Movie Reviews	Restaurant Reviews	Market Reviews	News Text
No. of Reviews	1,925	101	1,943	2,016	283
No. of Sentences	9,919	5,290	10,175	2,507	5,979
No. of Words	136,531	57,575	80,954	13,738	55,151
Avg. of sentences/review	5.6	50.8	5.2	1.3	20.5
Avg. of words/review	70.9	570.1	41.7	6.8	194.9
Avg. of words/sentence	13.7	10.6	7.8	5.4	9.3
Inter-agreement	0.8	0.73	0.72	0.84	0.78

keys: "No." stands for number, "Avg." stands for average, "Inter-agreement" shows the agreement in the sentiment annotation process between the annotators.

Abdul-Mageed et al., 2011). It also demonstrates that the two annotators have a good level of agreement. To make this process go faster, the rest of the dataset was divided into two parts, and the annotators each worked on them separately. After the annotation process had been completed, the data was organized into folders and text files. The corpus is available freely for researchers.

3.4. Corpus Distribution

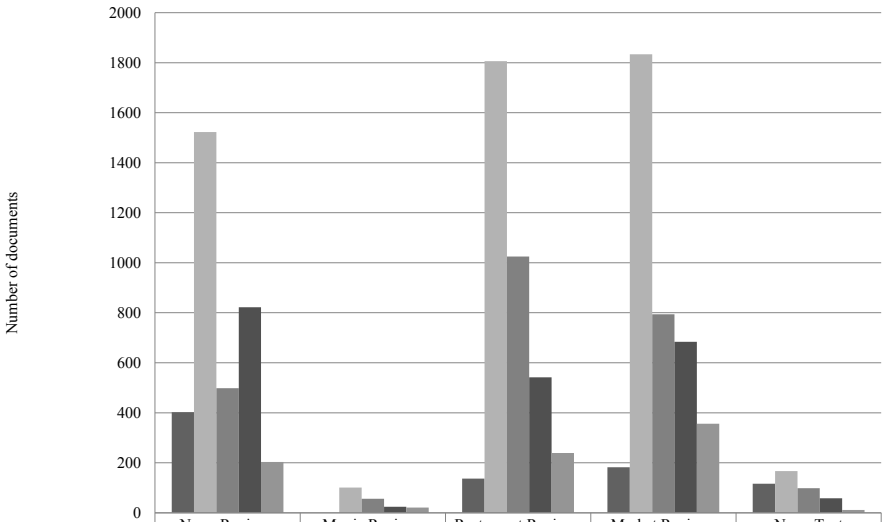
This corpus contains different types of the Arabic language. Modern Standard Arabic (MSA) is mainly applied in the news dataset and the movie review. Dialect Arabic (DA) is used alongside MSA in the rest of the dataset that includes news reviews, restaurant reviews, and market reviews. The dialect type is the Gulf and Hejazi Arabic dialect.

Table 3.1 shows the information about each dataset. As an example, the news review domain contains 1,925 reviews, and there is a total of 9,919 sentences, with 136,531 words in total. Some averages also are calculated. With the same example of news reviews, it can be seen that the average in a review is 5.6 sentences and 70.9 words. The average number of words in a sentence is 13.6.

From these numbers, it can be seen that there are three categories of reviews: long, medium, and short. The movie reviews tend to be long because they are written by critics in the movie field who express their feelings about the movie along with telling something of the plot and a number of facts. The same scenario also applies to the news text domain, which conveys the factual information about the news along with a few sentences concerning feelings. Restaurant and news reviews tend to be medium reviews, with approximately 5.5 sentences. This is due to the fact that the user tends to express feelings directly without adding factual information. The market reviews are small, with around 1.25 sentences. This is because the Arab user tends to voice their feelings simply about the market with few sentences. In addition, it might arise from the absence of punctuation because dialect Arabic is usually written without using punctuation marks to separate sentences. It is customary to see long sentences in Arabic that contain more than one idea and which could be divided into multiple sentences, particularly in DA.

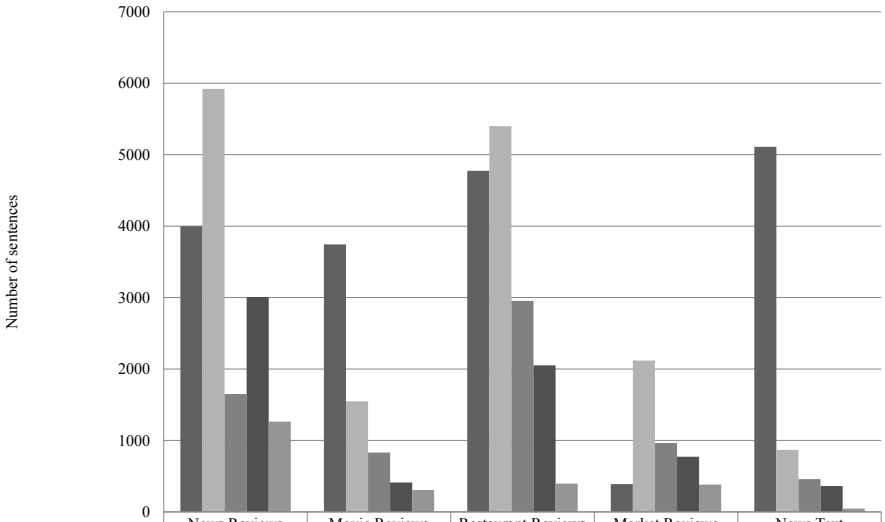
Figure 3.1 and 3.2 illustrates the subjectivity and polarity at the document and sentence level for each dataset. For example, the movie review does not have any objective reviews because all documents in this domain are subjective. There are, however, 56 positive, 24 negative, and 21 neutral documents. It is clear that the majority of the subjective documents over all the dataset are positive except the news reviews.

From these figures, we may infer that the subjective documents are around double those of the objective documents in most domains, apart from the news text domain. This may be because the primary aim of the review is to convey sentiment or feelings, whereas, that of the news is to convey information. In addition, three of the datasets (news reviews, restaurant reviews, and market reviews) have more subjective sentences than objective, due to the fact



	News Reviews	Movie Reviews	Restaurant Reviews	Market Reviews	News Text
Objective Reviews	402	0	137	182	116
Subjective Reviews	1523	101	1806	1834	167
Positive Reviews	498	56	1025	794	98
Negative Reviews	822	24	542	684	58
Neutral Reviews	203	21	239	356	11

Figure 3.1: Subjectivity and polarity at the review level for each dataset



	News Reviews	Movie Reviews	Restaurant Reviews	Market Reviews	News Text
Objective Sentences	3999	3743	4775	389	5110
Subjective Sentences	5920	1547	5400	2118	869
Positive Sentences	1649	830	2953	964	459
Negative Sentences	3008	411	2051	772	363
Neutral Sentences	1263	306	396	382	47

Figure 3.2: Subjectivity and polarity of the dataset at the sentence level

that their primary objective is to express feelings. In the news text and movie reviews, there are more objective sentences. This may be because the writer adds more facts and information about the term itself rather than putting forward his/her feelings. In the movie reviews, the critic or editor takes over five sentences (that do not contain any feelings) to describe the plots.

3.5. Removing noise from Arabic Data

The raw data is never 100% pure, and it may have some noise. The Arabic data that is obtained online includes non-Arabic words, special characters, Arabic words that have elongation (letter repetition) and symbols, non-Arabic characters, and numbers. First of all, all non-Arabic characters that may belong to HTML, links, or the programming language code have been removed from the text. The second treatment deals with special characters. These characters are sometimes used to express some emotion and sentiment such as smiley faces ":)", whereas the other types are used either by mistake or without an obvious reason. Whenever the special character is used to express sentiment, it was left with the text. If the character has not had any meaning, then it was removed from the text. To complete this process, the emotion special character list has been built to be used as a guide during the treatment process of special characters. The numbers also sometimes express a kind of sentiment or feeling, so, they are included within the text.

The user sometimes repeats some letters in a words as well as some digit in a number. A example is this sentence: "I like this story verrry much". The user tries to increase its feeling by repeating the letter "r" in the word "very" in order to express that he/she really likes that story. This behavior is also done in Arabic language. At the same time, we cannot leave the words as they are because this might affect the processing phase of getting the linguistic

Table 3.2: An example of word elongation in Arabic language

	without any elongation	With 5-times elongation
Arabic Sentence	مرحبا بكم	مـــــــرحبا بكم
Transliteration	$mrHbAbkm$	
English Translation	Welcome	

properties about the text such as, the morphology analyzer. In addition, the Arabic language has another unique feature. Elongation is a process of extending the shape of the letter in the words by using this special symbol "-". Table 3.2 illustrates an example of elongation in Arabic words. Bot of these issues (letter repetition and elongation) in Arabic language need to be resolved in the Arabic text before applying them to the morphology analyzer. For this reason, if the character is repeated three times or more, then they are made one letter. In addition, whenever the elongation is found, it was removed.

3.6. Arabic Morphology Library Package

To obtain necessary linguistic information from Arabic text, a part-of-speech tagger (POS) and Base Phrase Chunking (BPC), we used AMIRA (Diab, 2009) is used. This is a suite of tools for processing Arabic morphology. It is written using the Perl language. This toolkit includes different steps and tasks, comprising a tokenizer, POS and BPC.

The tools take a piece of regular Arabic text, process it and produce three different files. The first file includes the tokenization part, which tokenizes the Arabic words into the prefix/suffix/affix and the base words. The second file contains the POS for each word in the text. The POS scheme used was the extended tag set, which encodes some other morphological information, such as gender and number. The last file contains the BPC. The BPC is a process of dividing the sentence into phrases: noun, verb, adverb, adjectiv, or prepositional phrases. The BPC is considered to be a shallow syntactic parser. Figure 3.3

shows an example of the BPC in the second row. This sentence consists of six phrases. Each phrase contains one or more words. The noun phrase in this example contains three words. Each word in the phrase has its POS tag.

The AMIRA tool build depends on support vector machines in a sequence modeling framework, using the YAMCHA toolkit[5]. It is trained on MSA, but Diab (2009) claimed that this may also work for DA. Therefore, it was decided to use this tool for the dataset in the current study, as both MSA and DA were being employed.

In our corpus, each document and sentence has been tagged with a unique ID that might be helpful during the sentiment analysis process. In the case of the document, each of them has an ID in the first line that contains a unique identifier, its sentiment tags, and the number of sentences. Each sentence ID has information about the location of the sentence in the document as well as its sentiment tagging and unique ID. Figure 3.3 displays an example the sentence IDs. These IDs have been located in the first part of the document or the sentence because we do not want the AMIRA tool to process these IDs. Following these IDs, the Arabic text having been treated by AMIRA tool appears. This tool creates two different files, one for POS tagging, and one for BPC tagging. Figure 3.3 illustrates an example of these files. The first row explains the output of the POS tagger; the second row displays the BPC output.

3.7. Chapter Summary

This chapter focuses on explaining the details of how the Arabic sentiment corpus was built. The first step was collecting the data from different domains. It is then followed by the techniques of how the data is annotated. The details about the corpus are displayed by

[5]http://chasen.org/ taku/software/YamCha/

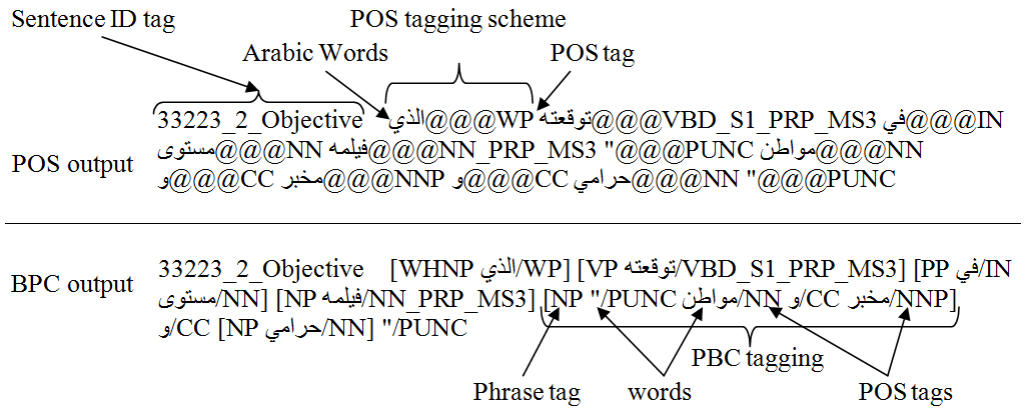


Figure 3.3: An example of morphology files that are generated from AMIRA

stating some of the distributions about the data nature in this corpus. At the end of this chapter, some of the unwanted data are noticed in the described corpus and their removal described. In addition, AMIRA, which is the morphology library, is used to build linguistic features of the data of our corpus. This next chapter starts illustrating the method of applying machine learning methods to Arabic sentiment analysis. In addition, it describes the different features that have been used to build feature vector model which is used to train the classifier algorithm.

CHAPTER 4

Feature Engineering and Machine Learning Classifiers

4.1. Introduction

As explained previously, getting sentiment from text can be solved with two approaches, the semantic and machine learning based approaches. Our work relies on using the machine learning approach to resolve the Arabic sentiment problem. In the English language, this issue has been investigated thoroughly, whereas the studies in the Arabic language are still in the beginning stages.

Using the machine learning "ML" to analyze sentiment from text requires a number of different steps. The first step is to have an annotated corpus for the data. In our case, we have built our Arabic sentiment corpus, and explained the details about it in the previous chapter. After that, the text needs to be converted to a model suitable for ML algorithms. This model is called the vector model or feature model, that consists of a matrix of numbers. Each column of this matrix represents a unique word in the corpus. Each row represents the document or the sentence depending on the level of classification. The value for each row and column displays the frequency of the word in the document or the sentence. While building this model, some of the unique features might be chosen or generated to be added to the matrix. After building this model, the ML classifier is trained with some data and tested with the remaining data to measure its performance.

Noted in Chapter 2, there is little research done in Arabic sentiment analysis. Some of the works only consider one type of level classification such as document level. The other

works include only two types of polarity (Positive and Negative) in polarity classification. In addition, most of them only work on one type of Arabic language, MSA, and do not investigate Dialect. Finally, most work has been done only on specific domains, mainly news or movie reviews. Therefore, we will try to investigate and evaluate Arabic sentiment analysis in different types of classifications, considering neutral as a polarity class, two levels of classification (document and sentence). We will also do this among different types of data and Arabic language forms. This evaluation will be achieved by proposing different feature sets and using different machine learning classifiers.

This chapter describes the work that has been done in choosing different features to build vector models and using different machine learning classifiers to analyze sentiment in Arabic text. The first part of this chapter shows the general methodology that we follow to achieve our goal. This methodology includes the method of how we build our features sets of Arabic text as well as the ML techniques that are used during the classification process. The following section shows the basic proposed features that have been used followed by the advanced proposed features. For each of these sections, the details of the method behind the proposed features will be discussed, followed by experiments that have been done.

4.2. Arabic Sentiment Analysis Methodology

4.2.1. Overview of our Approach. This section describes the method we used to process Arabic text for sentiment analysis. It also illustrates the tools that are used to classify the subjectivity and the polarity of Arabic text. Experimentation are conducted at two levels (document and sentence), in five different domains, and using different ML classifiers. This chapter starts with an overview of the steps that are followed. After that, the machine learning classifiers are briefly described.

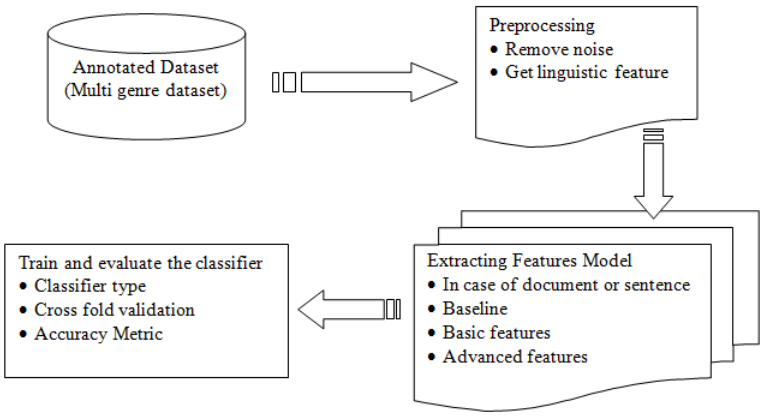


Figure 4.1: Work flow and steps in Arabic sentiment analysis

In the machine learning based approach, sentiment analysis needs an annotated corpus. The second step is preprocessing the data in order to eliminate any unnecessary data (noise data) from the original data or to prepare the data for the next step. The details of this step will be discussed in the following section. After the data is ready, the feature representation is constructed. This step is crucial because how the features are chosen and built will affect the performance of the ML classifier algorithm. The next step is to train and evaluate the classifier on the selected feature set. Figure 4.1 illustrates the steps of this approach.

4.3. Machine Learning Techniques

In this section, the theoretical foundations of machine learning algorithms that are used during classification are briefly described. Naive Bayes (NBs) is considered to comprise a simple model that works well on text categorization (Lewis, 1998). The Multinomial Naive Bayes (MNB) model was used for our experiments as it works better with word appearance in the documents (McCallum et al., 1998). Support Vector Machines SVM are one of the powerful ML algorithms that are used to solve classification problems. They work by assigning data to one of two disjointed half-spaces in either the original input space of the

problem for linear classifiers or in a higher dimensional feature space for nonlinear classifiers (Pang et al., 2002). These ML classifiers will be used in the basic experiments.

4.3.1. NAÏVE BAYES. The Naïve Bayes classifier has been used in the document classification problem for decades (Segaran, 2007). In addition, it is used as a baseline method to compare the performance of new methods (Jurafsky and Martin, 2000). The assumption of this classifier is independence between every pair of features. The Bayes Rule is the integral part of all Bayesian Models. This rule can be calculated as follows:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)},$$

where: $P(H|E)$ is the posterior probability of the hypothesis, $P(H)$ is the prior probability of hypothesis, $P(E)$ is the prior probability of Evidence, and $P(E|H)$ is the conditional probability of Evidence given Hypothesis (likelihood).

In our case, there are two hypotheses or three hypotheses depending upon the number of classes that we have. In the subjectivity classification, there are only two hypotheses, and we would choose the one that has the highest probability to represent the category of the text being either subjective or objective. Each text, either document or sentence, is represented as a feature vector. The H refers to the class (c) that we have, and E refers to the text (t) that we have. We get the following equation from that:

$$P(c|t) = \frac{\Pi_{i=1}^{F} P(f_i|c)P(c)}{P(t)},$$

where F is the total number of features, and $\Pi_{i=1}^{F} P(f_i|c) = P(t|c)$.

Recently, Multinomial Naïve Bayes (MNB) has been demonstrated in text retrieval problems. The MNB is another type of Bayesian classifier that relies on Bayesian Rule. In Naïve Bayes the normal or the Gaussian distribution are used for each feature, whereas the Multinomial distribution is used in case of MNB.

4.3.2. Support Vector Machine. A Support Vector Machine (SVM) is a type of supervised learning technique that is used to analyze, recognize, and classify data (Bishop, 2006). The SVM is a discriminative model that is used widely in different types of machine learning concepts, such as linear and nonlinear regression, as well as classification (Bishop, 2006).

SVM belong to the general category of kernel methods (Bishop, 2006). The SVM projects data points in higher dimensions in order to make the data points linearly separable by using the kernel techniques. There are different types of kernels that might be used. In our work, we will use the linear kernel because it is the state-of-the-art technique that has been used in sentiment analysis (Pang and Lee, 2008). We also compare the nonlinear kernel (polynomial kernel) with the linear.

The basic concept of SVM is that it is looking for the Optimal Separating Hyperplane called w between the two classes by maximizing the margin between the classes' closest points (see Figure 4.2). The points are located on the boundaries are called support vectors. The middle of the margin is our Optimal Separating Hyperplane. The SVM classifier tries to find the optimal hyperplane among the possible hyperplanes between the different classes.

4.3.3. Machine Learning Library. In this work, we relied on the scikit-learn library (Pedregosa et al., 2011) for using different machine learning classifiers. This library is an open-source machine learning library for the Python programming language. It includes

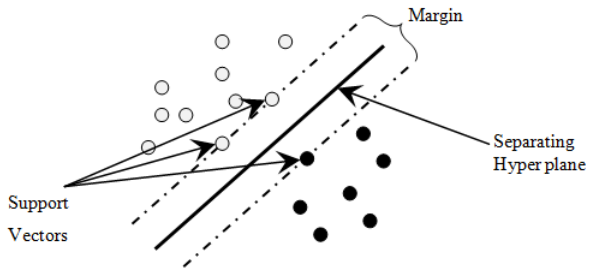


Figure 4.2: Support vector machine concept

classifiers, such as Naive Bayes (NB), a Support Vector Machine (SVM), logistic regression and other classifiers. In addition, it provides an easy way to use the ML classifier, which makes scikit-learn more user-friendly. Finally, the performance of the library is faster than that of other libraries that implement ML algorithms (Pedregosa et al., 2011).

4.4. Preprocessing

Preprocessing has been explained in the Chapter, 3. However, a brief description of this stage will be shown here for the sake of clarity. The preprocessing phase contains four main steps before the documents or sentences pass to the classifier in a vector model form. The first step includes the filtering out of all rubbish data that might be found in the text, including single letters or non-Arabic characters. The second step is to normalize long words that may make some letters redundant. For example, some users tend to write (مشكووووور / *mškwwwwwwwr*/ Thankssssss). The normalization here will reduce the repeated character to one letter only: (مشكور / *mškwr* / Thanks). The third step is to use the AMIRA toolkit for all data, in order to prepare the POS tag for all words. The final step involves removing the words that belong to the stop word list. We added the Arabic stop words list (El-Khair, 2006)to the scikit-learn tool. This will help to remove the stop words while the tools builds the vector model that represents the Arabic text.

Table 4.1: Representation of the feature vector model

	w_1	w_2	w_3	... ...	w_j
D_1	1	0	1	... ...	0
D_2	0	0	1	... ...	1
...	...	...	...	... ...	...
D_i	1	1	0	... ...	1

Key: "D" refers to the document, and "w" indicates the word

4.5. Feature Space Model Preliminary

Machine learning provides many algorithms that work for classification, but the challenge of finding a sentiment in a text is determining the best feature to be used. The following sections reveal the common features that are used in sentiment analysis.

4.5.1. Feature Model. Before using a ML classifier on the data, we need to represent the text in a format suitable for the classifier to deal with it. In NLP, the popular model is the vector model or feature model. The text, either document or sentence, will be converted into the form of the features model before the training process of the classifier starts. This model should preserve essential information about the text. Each row of the model represents one of the data set records (either document or sentence). Each column displays the features that are chosen to build the vector model. The intersection of each row with each column contains a value that represents the relation of that feature in that data record. Table 4.1 illustrates an example of a feature model using the Bag of Words (BOW) as a feature to the text. In this model, the features are the distinct words of all text in the corpus. Each column represents one word from the dictionary that is built of all the distinct words in the corpus. The rows correspond to the documents or the sentences of the corpus. The values in the table show if the word occurs in the text (document of sentence) or not. The next section explains the details of this value.

4.5.2. Term Frequency Versus Present. Term Frequency is the measurement of how many times a particular term is repeated in a document. This has long been emphasized in traditional information retrieval systems. Term presence is another model that shows the existence of the term in the document in a binary mode. The document model here shows that term presence is 1 if the term appears at least once in a document, and 0 if not. This model is used in (Pang et al., 2002) and shows improvement compared with the term frequency model. The most famous model in the field of NLP is the one that uses term frequency and decreases the effect of the high-frequency term by using the inverse document frequency (TF-IDF) (Sparck Jones, 1988). The IDF determines whether the term is common or rare in all the documents. In addition, the word appearance is very informative when compared with the word frequency (Pang and Lee, 2008). Therefore, the term appearance, or word presence (TP) was used to build the feature model in this work. The presence model is chose in this work because it has been the most useful model that is used in sentiment analysis field (Pang et al., 2002; Pang and Lee, 2008).

4.6. Feature Design

This section explains the features that we use during our experiments. We refer to some of the features as Primary Features. Moreover, we propose to use new features that might help in sentiment classification. These types of features are designated Advanced Features.

4.6.1. Primary Features for Arabic Sentiment Analysis. This section shows the details about the primary features that are used during the building of the feature vector model. These features are categorized as informative, semantic, or stylistic.

4.6.1.1. *Bag-of-words Feature.* The bag-of-words (BOW) feature is sometimes called an n-gram model. In this type of feature, the basic informative aspects of the text will be

preserved and used to build the feature vector model. This feature consists of the distinct words in the corpus in different n-gram models. In the case of the uni-gram model, the feature of BOW will contain only one word from the distinct words of the corpus. There are some variations of this model which include bi-gram, tri-gram, and etc. models. In each of them, the feature will contain a combination of two, three, or n words depending on the model type. For example, the feature column should consist of two words in the bi-gram model. In sentiment analysis, the positions of the term are significant in a document representation. Therefore, choosing a good n-gram model plays the central role of sentiment classification. The benefit of using n-grams might appear in being able to capture some dependencies between the words and the importance of individual phrases in sentiment.

4.6.1.2. *Part of speech.* This feature is considered a kind of semantic feature that might capture some of the linguistical features about the word. As previously explained, it is important to find the adjectives, as these are good indicators conveying the sentiment orientation in the text (Benamara et al., 2007). Using the part-of-speech "POS" tagging system decreases the ambiguity of the word (Wilks and Stevenson, 1998). When a word is annotated with its POS tag, it helps to increase the NLP system's confidence in its actual meaning. This will help significantly in the case of more morphological languages such as Arabic. For example, the word (**جمل**/ *jamal* or *jam ala*) could be the noun "camel" or the verb "make something beautiful". The POS tag feature will help to determine the correct meaning of the word. Turney (2002) also used the POS feature for adjectives and adverbs in order to obtain the sentiment orientation at document level.

In this direction two features, POS and Adv&Adj, are used. To build them, we first apply AMIRA to our data text to get POS for each word in a sentence. The first model

is the one that includes the POS tag of the words. This model is constructed by using the word along with its POS tag. For example, for the word "eat" if it is a verb, the POS tag V will be used. The word "eat" will be attached with its POS tag as follows: "eat_V". The second model, Adv&Adj, is concerned with only two types of POS, adjective and adverb. This model is only built using the adjective and adverb words that are found in the sentence. This will reduce the size of the feature model but might lose more information about the text.

4.6.1.3. *Stylistic Features.* Some other features, such as the style of the text, may contribute to the sentiment orientation of the text (Abbasi et al., 2008). For these types of features, we used some stylistic features that we hypothesized would play a role in Arabic sentiment analysis. As explained earlier, some research has shown the effect of using the length of sentences as a feature in sentiment analysis (Abbasi et al., 2008; Na et al., 2004). In the case of document classification, the number of sentences in a document will be used as a stylistic feature. The number of the words will be used in the case of sentence classification level.

4.6.2. Proposed Advanced Features for Arabic Sentiment Analysis. This section sheds light on the proposed features that we used in Arabic sentiment analysis. We refer to them as advanced features because they are either not used often with Arabic sentiment analysis or are utilized in another language such as English. Some of these proposed features might work in a document level classification, whereas the other features work in a sentence level.

4.6.2.1. *Position of the Sentiment.* For document level classification, it has been proposed that the use of some parts of the documents, especially in the case of a long document,

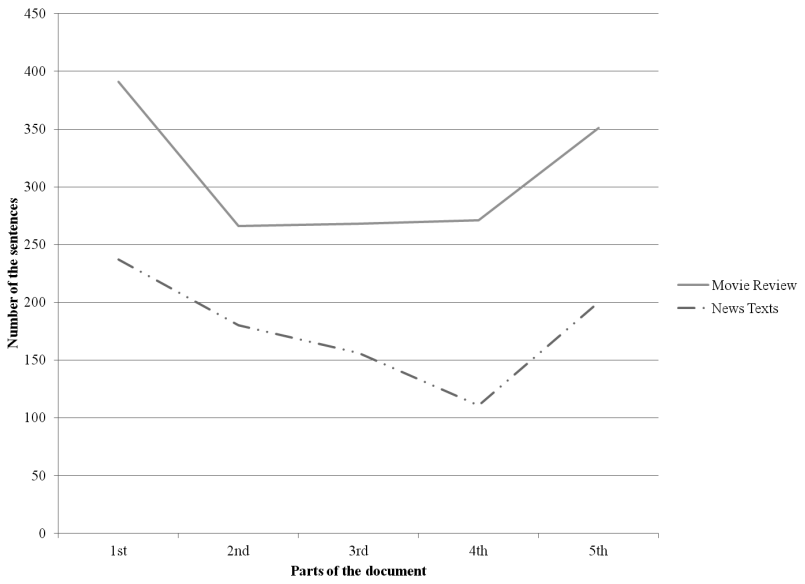


Figure 4.3: Number of opinion sentences depends on its position in two different domains

might improve the accuracy result for Arabic sentiment analysis. This intuition comes from the way in which users tend to give their opinions on a particular subject. Users express their feelings either at the beginning or the end of their writing. They almost always place some factual information, sometimes combined with opinions, in the middle of their writing. Therefore, we have investigated this for Arabic by counting the subjective sentences in different positions in each document. Only two domains are used for this proposed feature: the movie reviews and the news domains, as they are considered as long texts. In order to do this, each document is divided into five parts. The sentences of each document are distributed equally to each part. Then, the number of sentence in each sentiment class is counted as each part of the document. For example, all positive sentences will be counted for the part 1, part 2, and so on for each other subjective category. Figure 4.3 shows the results for the News text and Movie reviews.

The results suggest that the most "feeling" opinioned sentences occur in either the early or the late positions of the document. Therefore, if only the first and the last parts of the

documents are considered to analyze the sentiment in long Arabic texts, this method may reduce other noise that could interfere with the actual feeling of the writer. It seems that most of the objective sentences occur in the middle of the document which may add some noise to the classification, especially in polarity classification. This is one of the proposed features that are investigated in our document level classification experiments.

4.6.2.2. *Base Phrase Chunk.* The Base Phrase Chunk in NLP application is the process of finding the logical phrases of the text. In the linguistic field, there are many phrases that might build the sentence as what we have in the Part-of-speech (POS) of the word. For example, a verb phrase should contain the subject and the object that did the action of that verb. An example of this concept is explained in detail in Chapter 3. In addition, the BPC is considered as type of shallow dependency tree of the text that shows the relation between different words in a sentence (Diab, 2009).

The BPC are generated during the preparation of our corpus as explained in Chapter 3. To build this feature, we apply the same mechanism that we used to the POS feature. The word will be attached with a phrase tag where it is located. For example, say we have some words $\{w_1, w_2, ..wi\}$ which are situated on the verb phrase part $\{V - PHR\}$. These words should have the verb phrase tag beside them during the build of the feature model. This process generates the following feature: $\{\ w1 - V_PHR,\ w2\text{–}V_PHR,\ ..\,wi - V_PHR\}$. In addition, we also add all phrase chunks as it is in the feature model. Each phrase of the sentence is used as a feature in the model. Let's a definite sentence has three phrases: noun, verb, and adverb phrases. Each of these phrases is used as new feature and adds them to the feature model. That means we will capture the effect of the phrase in sentiment analysis.

4.6.3. Word Polarity Score. This proposed feature relates to the semantic orientation approach of the sentiment analysis techniques. Relying only on the actual word to build a feature model is a good starting point in sentiment analysis but there is a need to add more information about the text in the feature model. The semantic orientation technique in sentiment analysis only relies on calculating the polarity score of each word in the document. After that, the final decision about the text's sentiment is taken depending on the calculated value. When the polarity score is added to the feature model, this approach may get the benefit of semantic technique and merge it with the ML method. Therefore, adding this feature results in a hybrid method when the ML technique is used as a primary classifier and supports it with some of semantic orientation concept.

In order to obtain the value of word polarity, a lexical resource, such as SentiWordNet, is needed. This lexicon is constructed from the perspective of WordNet to which each synset, which is a set of one group of synonyms, is assigned three sentiment scores: positivity, negativity, and objectivity. In the Arabic language, there is a lack of these resources, which are either not available for free or are incomplete. Abdul-Mageed, et al., (2011) manually built an Arabic lexicon comprising a list of approximately 4,000 Arabic adjectives from the newswire domain and annotated for polarity. This corpus only contains one type of POS, adjectives, and is not comparable with the English SentiWordNet. It is only a collection of the positive and negative words without any of the scoring values. Recently, Alhazmi, et al., (2013) discussed the issue of building the Arabic SentiWordNet and started to put the first step in place to create this corpus. However, they are still working on it in order to enhance its performance before making it free publicly.

Until the time of preparing this work, there had been no available Arabic SentiWordNet. Therefore, we relied on the English SentiWordNet by first using machine translation to convert Arabic words to English. Ghorbel and Jacot (2011) found that using a machine translation to obtain the polarity score in the French language improved the performance of sentiment analysis.

Figure 4.4 shows how the polarity approach might be injected into Arabic sentiment analysis. It combines with the fundamental approach that we applied in our primary experiment. The difference is the polarity score calculation. This part is responsible for calculating the score of a given word. Figure 4.5 illustrates the details of how the polarity component works with Arabic sentiment analysis. In order to calculate the polarity score, we have to have SentiWordNet. In our case, this lexicon does not exist. Therefore, we rely on an alternative approach. We believe that the optimal solution is the one that has a native Arabic SentiWordNet. However, relying on a mature SentiWordNet in another language and use Machine Translation Mechanisms might help to evaluate this approach.

In order to build and use this approach with Arabic sentiment analysis processing, we downloaded the latest version of the English SentiWordNet[1]. The other part of this component is the translation unit. We rely on the translation service provided by Google translation API to implement this translation unit. The polarity score then is calculated for each record while building the feature model. In the case of document classification, for each document the total polarity score should be calculated. This leads to four feature columns in the space model: one for positive, negative, neutral, and objective. For example, say we have document d that has 40 words, our method translates each of document's words and gets its

[1]http://sentiwordnet.isti.cnr.it/

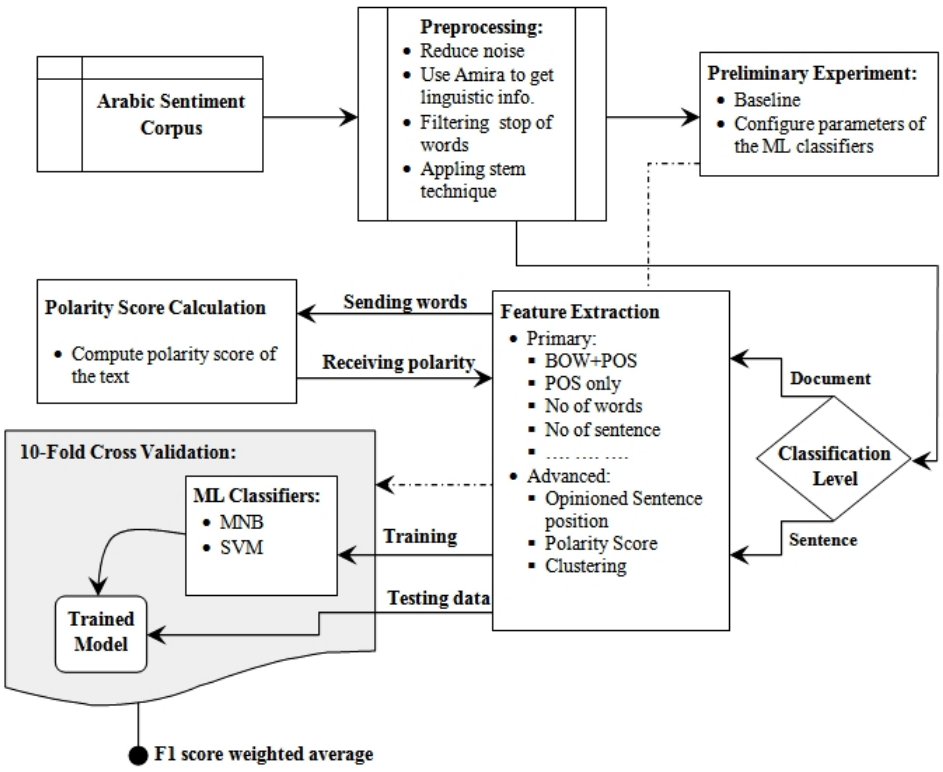


Figure 4.4: Adding score polarity to Arabic sentiment analysis

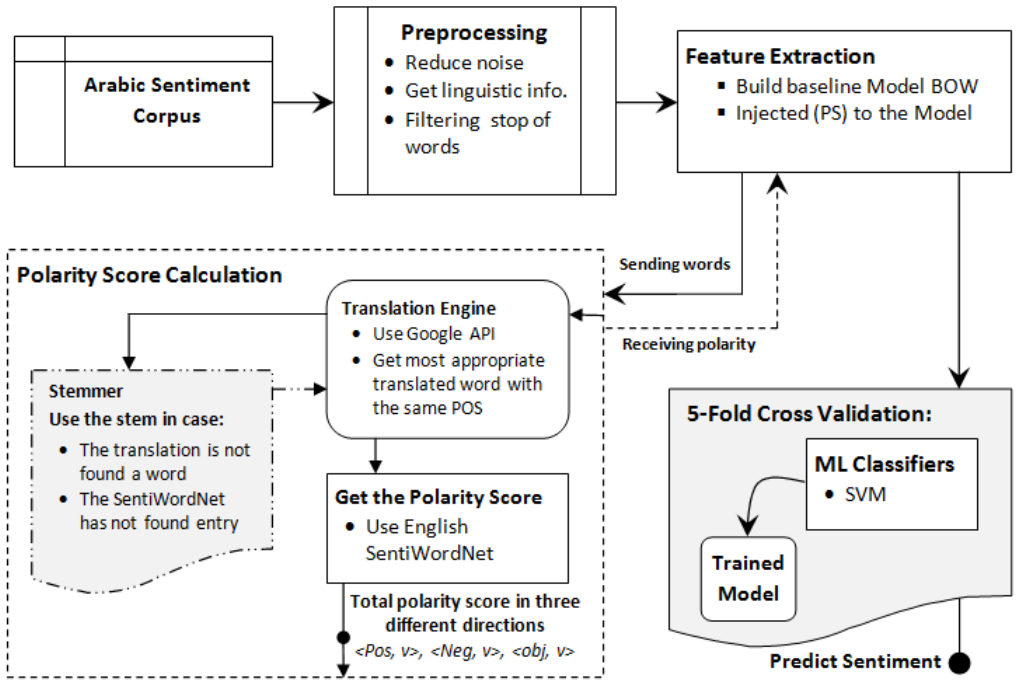


Figure 4.5: Illustrates the details of the polarity score calculation component

Table 4.2: Number of words that have or do not have translated pairs with different stem methods

	No Stem	With Stem	Stem First
News Reviews	65.5%	85.1%	74.0%
Restaurant Reviews	64.5%	83.7%	72.4%
Market Reviews	69.4%	86.1%	75.3%
Movie Reviews	79.7%	91.8%	78.3%
News Text	81.5%	91.8%	79.1%

Key: "*No Stem*" indicate no stem is used during the translation process, and "*With Stem*" shows that the stem is used if the actual word have not translated word, and "*Stem First*" displays that the stem is applied first and then the translation is performed.

score from the English SentiWordNet. The final total of the four features will be calculated and the end. In the end, we have the following features for document d:

$$d : [positive, value], [negative, value], [neutral, value], [objective, value]$$

To make a comprehensive investigation of this approach, we vary between choosing what polarity score we add in each classification types. For example, we only consider the positive and the negative score of polar words during the polarity_2 classification that includes positive and negative text.

Table 4.2 shows the percentage of words that have been translated in each domain of our dataset. The first column illustrates the proportion of the words that have been translated. For example, 79.7% of the words in the movie reviews can be translated to the English language. In addition, we notice that there is a difference in the percentage of translated words among the different domains. We can divide this data into the ones having a high percentage of translated words and those with low percentage of translation words. In the high one, we can notice that both of these data sets use MSA, whereas the low one has DA in most of their parts.

This issue lead us to use other techniques with this approach to help to reduce the number of untranslated word. The stem mechanism is used to minimize the variation of the word and

may preserve the semantic meaning of the word (Farghaly and Shaalan, 2009). Therefore, the stem mechanism is applied to the word before the translation.

In the literature, there are three different root libraries for the Arabic language: Khoja Arabic stemmer (Khoja and Garside, 1999), ISRI stemmer (Taghva et al., 2005) and Tashaphyne Light Arabic stemmer Tashaphyne (2010). The most suitable root library is Tashaphyne Tashaphyne (2010) because it has a real implementation using Python and can be used with the other tools that are utilized in the classification process.

For this feature, the word is first translated. When there is no translation found, it then transfers to apply the stem to it. After that, the stem word is translated. In addition, applying this method should contribute to evaluating whether the root mechanism preserves the sentiment orientation of the actual words or not. Table 4.2 illustrates how the numbers of un-translated words reduce by applying the root mechanism in the second column. The percentage of the words translated increased by around 20% using this method. Moreover, we investigated using the stem directly before using the translation. We found that the percentage of translated words was better using the stem before the translation than the first approach that does not use the stem, but is not better than the second one using the stem after the translation. The percentage of the words that are not found in the SentiWrodNet was small, which is around 2.2% of the total number of words in all domains.

4.6.3.1. *Polarity Counting Versus Scoring.* We rely on two different mechanisms in this proposed feature. The first one depends on computing the score of the polarity of the text. As explained earlier, the polarity would be calculated in the particular text for all words in that text. This score would be categorized into four different types: positive, negative, neutral, and objective polarity. The second mechanism is the one that relies on just counting

Algorithm 1 The steps of calculating polarity

- For each words W in the text T do the following:
  - Translate the actual word using translation engine: W would be TW
  - If the w is not translated, translate the stem word to set TW
  - Get the polarity score of TW from English SentiWordNet The output would be {Positive: S, Negative: S, Objective: S}
  - Determine the polarity type of TW and its score:
    - If the Positive score > negative and Positive > = objective , then TW is positive with that score
    - Else If the Negative score > positive and negative > = objective, then TW is negative with that score
    - Else If the Objective score > positive and Objective > negative, then TW is objective with that score
    - Else If the positive score = negative and positive > objective, TW is neutral with that score
  - **Compute the Polarity (Counting or Score)** as follows:
    - Let assume that:
    - Text T has words vector W $\{w_1, \ldots, w_i\}$
    - we have four different polarity types PT{positive, negative, neutral, objective}
    - for every polarity in PT:
    - $$PolarityCount = \sum_{i=1}^{n} PolarityType(w_i),$$
    - $$PolarityScore = \sum_{i=1}^{n} Polarity(w_i),$$
    - Where w represents all words in the text and *PolarityType* is the function that return polarity of the word
    - and *Polarity* is the function that calculates polarity score of the word

the number of polar words. For each type of polarity, we count the words in each sentence. For particular texts, the polarity of each word should be known through the SentiWordNet, and we count the number of positive, negative, and objective words that we found in each text. The algorithm 1 explains the pseudo code of our approach.

Some research uses a similar idea to our approach in Arabic language (Abdul-Mageed and Diab, 2012a,b). However, our approach is different from the other in various aspects. The first one, we use the actual polarity score instead of only using the polarity words and

build the feature depending on that. Abdul-Mageed, et al.,(2012b) started to build Arabic lexical resource from different domains. Their lexicon only contains the actual words and classifies them into different polarity category. That means this corpus has only the words without any scoring value. After that, they use this corpus as a dictionary to build the feature model instead of using all words in the actual dataset of the main sentiment corpus. In our approach, we rely on the value instead of the word only, and we include all other word in the text that might carry the sentiment orientation of the text.

4.6.4. Using Word Clustering as Feature. The last proposed feature is the Word Clustering. Word Clustering is a process used to distribute words that have the same semantic or syntactic relationship within the same group. After the clustering process is complete, the Cluster Label of the word is used as a feature. This feature achieves better performance in different Natural Language Processing (NLP) tasks, such as Name Entity Recognition (Tkachenko and Simanovsky, 2012). This feature may support the classifier in capturing the similarity between words and the sentiment orientation of the words. In addition, this may be useful in the case of DA when there is a lack of morphology tools that work well with this type of Arabic language.

Many of the NLP applications rely on the BOW model representation and some other features, as explained earlier. In some languages that are considered high inflectional language such as Arabic, this BOW model is very sparse due to the richness of the vocabulary even after using the stem technique (Habernal et al., 2014). In order to tackle this issue of the model, we enriched the baseline model with the word cluster tag. This comes from the assumption that the words that are in the same cluster are semantically substitutable. In

addition, we investigated if the word cluster might also preserve the sentiment orientation of the words.

In order to perform this proposed method, we need to group the words into different clusters. The first step is to use a suitable clustering algorithm that works well with the Arabic language. Among different word clustering algorithms such as Reduce Dimensionality (Collobert and Weston, 2008; Mnih and Hinton, 2009), and distributed word embedding (Lamar et al., 2010), the Brown clustering words algorithm has been used as a standard technique in the many NLP problem (Liang, 2005). This clustering is used due to the simplicity and hierarchical nature of its output and the implementation availability. Therefore, we will use this algorithm in our work.

4.6.4.1. *Overview of Brown Clustering.* This clustering algorithm is considered as the class-based bi-gram language model. It works by maximizing the mutual information of adjacent clusters (Brown et al., 1992; Liang, 2005). The central idea of this cluster is grouping words that have the same distribution as neighbor words. This algorithm tries to cluster the words depending on their context in the same data. It takes a word and computes the probabilities of this word occurring in a similar context. For example, this cluster would learn the probability distribution of neighbor words of the words, such as Jeddah, to be similar to neighbor words with another word like Denver. This intuition can be inferred from these two words being the names of known locations, that is the names of two major cities. In addition, the clustering algorithm supposes that the context of these two words should be similar. By the same concept, we could assume that the sentimental words might come in the same context. As a result of that, the clustering algorithm would cluster them in one group.

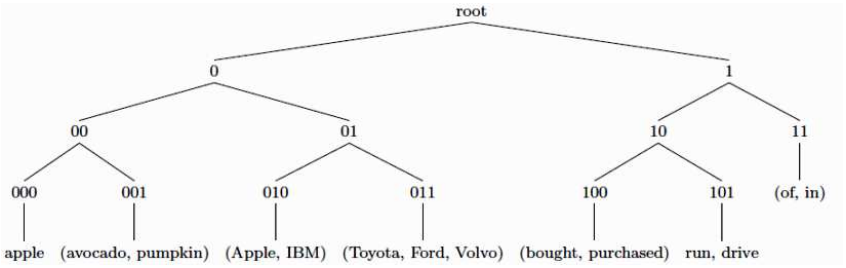


Figure 4.6: An example of Brown word clustering algorithm output

The Brown Cluster algorithm is a words cluster-based approach that takes a sequence of words $(w_1, w_2, \ldots, w_n)$ as an input and generates the cluster of those words as a binary tree. The leaf of this tree contains the words, and the internal nodes represent the cluster bit string. An example of the output of this clustering technique is shown in Figure 4.6.

Suppose we need to cluster some data into 50 cluster groups. At the end of the cluster algorithm process, it will generate 50 cluster names at the leaf of the tree. Each group might contain one or more words. Then, the clusters are grouped into one standard upper cluster in the binary fashion. This process generates until the root is reached. Figure 4.6, the words (bought and purchased) are grouped in one cluster. Their cluster tag is 100. This tag is called bit string ID. This ID starts from the root to the leaf. The sibling cluster that is 101 has two words that are run and drive. From this cluster, we could infer that the verbs in group 100 have a similar meaning, that refers to the business sense (buying). In the other cluster 101 the meaning is different from the (100) cluster. In the case of the upper cluster tag (internal node) which is cluster 10, the words would be all verbs in all sub-clusters belonging to the parent. This also represents syntactic or semantic features of these words. Notice that all of these words are verbs. More details of the algorithm specification are presented in (Brown et al., 1992).

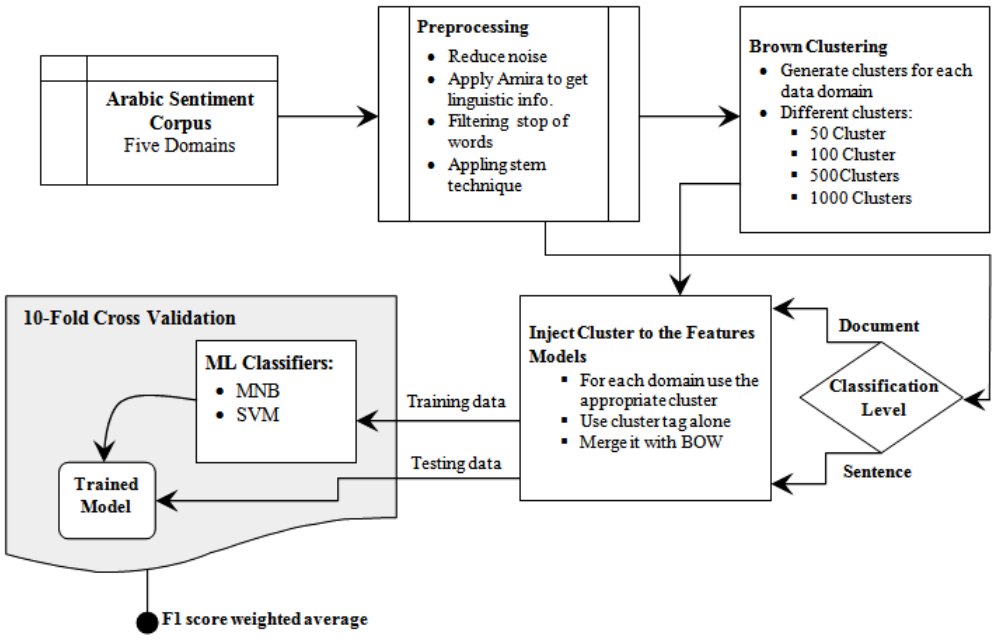


Figure 4.7: Word Cluster in Arabic sentiment analysis

4.6.4.2. *Injecting Word Clustering into Feature Model.* Figure 4.7 describes the steps of this experiment. Firstly, the clustering algorithm[2] processes all data in order to group all words into different clusters. After the clustering step is done, we will have a database of all words of our sentiment corpus with their cluster tag. The cluster tag indicates the cluster group that the word belongs to. We have four different number of clusters group for each data domain in our corpus. Each of our data domains is processed by the Brown cluster algorithm into four different cluster numbers (50, 100, 500, 1000). The typical cluster number that is used in research is 1000 clusters (Liang, 2005; Ratinov and Roth, 2009; Tkachenko and Simanovsky, 2012). We also use a small number of clusters to investigate the effect of that in the different types of domains that we have.

The second step in our proposed features determines how we can use this information during Arabic sentiment analysis. The first proposed approach is to use the cluster tag by

[2]We relied on the implementation of Liang (2005) for the Brown clustering algorithm

itself as a feature to build feature model. That means we only rely on the cluster label of the words to build the feature model. The second method is to inject this feature with the standard BOW model that is the baseline of our experiment. The word would be attached to its cluster in the feature model as what we follow in the POS feature. The last method is to combine the first model with the BOW model.

4.7. Experiments

This section illustrates the experiments that are done to investigate and test the features and performance of the ML classifier on Arabic sentiment analysis.

4.7.1. Experiment Setup. The experiments can be divided into two levels of classification: one on document classification and the second on sentence classification. The features suitable were used in each level. For example, the position of opinioned sentences is only used in document classification. The first step of our experiments started with the sentence classification and was followed by document classification.

In order to get fair and smooth evaluation performance all experiments are reported as the average of a 5-fold cross validation. N-fold cross validation is used in the majority of computational linguistic research because it is a reliable accuracy measurement method. In our case, we divide our dataset into five disjoint parts with equal proportions of samples in each class. Four of them are used to train the classifier while the rest will be used to test the model that is generated during the training process. That means the classifier will be trained on 80% of the data and used on 20% for testing. This process will be repeated five times because we have five partitions of the data. Every time a new partition is used for the testing phase. During every cycle, the F1 metric is calculated that measures the accuracy of the classifier. The next section will explain this metric. In the end, we will have five

Table 4.3: Confusion matrix

		Predicted Class		
		A	B	C
Actual Class	A	T_{AA}	F_{AB}	F_{AC}
	B	F_{BA}	T_{BB}	F_{BC}
	C	F_{CA}	F_{CB}	T_{CC}

F1 values for each fold, so the average of these values will be calculated in order to get one unified value that is used to evaluate the performance of the classifier on the selected features sets.

4.7.2. Evaluation Metric. To measure the performance of the classifier, the F1 score is used after computing the precision and recall. The accuracy shows the overall correctness of the model by averaging the correct classifications on the total number of classifications. The precision measures the accuracy of the classifier in regards to the specific predicted class. The recall is sometimes called the sensitivity of the classifier as it is the percentage of the correct predicted classes among the actual class in the data. Suppose there are three categories that the classifier is trained on. After the testing is done on the model, we will have a confusion matrix. This matrix shows the number of each correct and incorrect class items in each category. Table 4.3 displays an example of the confusion matrix. The symbols that are inside the table refer to the number of samples of data that are classified either correctly or incorrectly to particular class category. For example, T_{AA} (True Classified as A class) refers to the number of items that are true in class (A) category. In addition, the F_{BA} or F_{CA} (Falsely Classified as A classes) illustrates the number of samples or data that are classified as class A class but they should be in class B or C.

From the confusion matrix, the precision and the recall can be calculated as:

$$Precision_A = F_{AA}/(F_{AA} + F_{BA} + F_{CA})$$

$$Recall_A = F_{AA}/(F_{AA} + F_{AB} + F_{AC})$$

The average of precision and recall also is calculated for all classes. Then, the F1 score is calculated as:

$$F_1 = 2\,\frac{Precision \cdot Recall}{Precision + Recall}$$

After the F1 score is computed individually for each class, the weighted average of F1 is calculated to establish a single value that can be used to evaluate the performance of the classifier. For example, an MNB classifier is used to classify subjectivity, i.e., the document is either subjective or objective. The F1 score will be calculated for each class individually (F1 for the subjective and F1 for the objective). Finally, a weighted average of F1 is calculated, resulting in a single value. The weighted average is calculated as:

$$F_{1_{weighted\ average}} = \frac{\sum_{i=1}^{n} W_i \cdot f_i}{\sum_{i=1}^{n} W_i},$$

where f is the F_1 score for each class, and W is the numbers of documents or sentences that are used in the testing data in each class.

Though we have insufficient samples to do a meaningful test for statistically-significant differences, we did calculate and show the differences in the range of error of F_1 in each model, when it is applicable.

4.7.3. Baseline Experiment. One of the primary goals of this work is to evaluate the different feature sets as well as different ML classifiers for recognizing the subjectivity and sentiment of the Arabic text. In order to do that, we need to establish a proper baseline experiment before starting to do comparison experiments. This provides a useful method to compare the performance of different classifiers with the corresponding feature sets. The question here is what the best baseline is. It is hard to judge or make the optimal baseline

experiment because the baseline will vary depending on the nature of the task. In our case, we choose the simple features, the BOW uni-gram model, as the baseline that provides the point of reference for judging other feature set experiments for each classifier. This baseline might be fair because it preserves the basic knowledge about the text classification problem which is the general topic of sentiment analysis.

An important issue in working with ML classifier algorithms is how their parameters should be tuned. The different parameters values in the same classifier algorithm lead to the various testing results. Each of the ML classifiers that we used in this chapter or the following ones have multiple parameters that can be configured to change the algorithm's behavior. In the literature, the performance of the ML classifier might vary depending on the parameters chosen for the classifier in NLP problems (Daelemans et al., 2003). The optimal parameters might also vary according to the set of features are used in the space feature model (Daelemans et al., 2003). Although it is not the goal of this work to find the optimal parameters for each classifier for each feature setting, it is important to find a suitable parameter configuration for each classifier, in general. The parameters of the ML classifiers in this work have been selected during the preliminary and baseline experiments. To do that, we divide the data into three sets of partitions, training, validating, and testing set. The parameters are tuned during the training classifier and tested on the validation data. In the end, the proper parameter that is found during the previous process is then tested on the testing data. These chosen parameters and values are then applied to 5-fold cross validations.

4.8. Results and discussion

In this section, some of the experiments are performed at two different classification levels. The first groups of tests were carried out at the sentence level for various settings using two different ML classifiers. The second part of the experiments were at the document level. The main idea behind these experiments was to establish the best feature sets and ML classifiers that work well with each level of classification for sentiment in Arabic. In addition, among the goals of these experiments is to compare the features that might be suitable for both types of the Arabic language, that are Dialect Arabic (DA) and Modern Stander Arabic (MSA).

Our experiments in this section used two different ML classifiers to carry out our approach in Arabic sentiment analysis using the first set of features. The first classifier is Multinomial Naïve Byes that is referred by MNB in following sections. The other is Support Vector Machine (SVM) with the linear kernel. These two classifiers have been chosen because they are the state-of-the-art ML classifiers that are used in NLP and sentiment analysis field (Abbasi et al., 2008; Abdul-Mageed et al., 2011; Pang and Lee, 2008). The additional experiments are performed in order to investigate the performance of using a nonlinear kernel of the SVM with some feature sets. In all remaining experiments, the linear kernel is used because it shows the best performance compared to the other classifier.

In addition, all of our experiments are performed on different levels of classification and various types of classification. We refer to the sentence and document classification as a classification level. The types of classification include three categories. The first one is subjectivity classification, which determines whether the text, either document or sentence, is subjective or objective. The second type is two kinds polarity classification. This type

Table 4.4: The results of our baseline experiment at the sentence level

	Subjectivity		Polarity_2		Polarity_3	
	MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	67.4%	**69.2%**	57.5%	**58.1%**	55.5%	**57.3%**
Restaurant Reviews	70.2%	**71.0%**	81.0%	**83.4%**	72.1%	**73.2%**
Market Reviews	88.4%	**89.3%**	87.2%	**88.2%**	69.3%	**69.4%**
Movie Reviews	44.3%	**45.0%**	77.1%	**80.0%**	49.3%	**52.1%**
News	33.3%	**35.2%**	**82.1%**	80.1%	**73.5%**	71.2%

Key: "*MNB*" indicate the F1-score of using Multinomial Naive Bayes, and "*SVM*" shows the results using Support Vector Machine.

Table 4.5: The results of our baseline experiment at the document level

	Subjectivity		Polarity_2		Polarity_3	
	MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	86.2%	**88.1%**	54.1%	**56.4%**	**61.0%**	58.1%
Restaurant Reviews	95.0%	**96.2%**	84.2%	**85.3%**	65.5%	**67.0%**
Market Reviews	92.1%	**93.4%**	88.4%	**90.0%**	69.3%	**70.0%**
Movie Reviews	NA	NA	78.2%	**80.0%**	**49.0%**	44.5%
News	57.5%	**63.4%**	**77.1%**	76.4%	**69.0%**	65.3%

Key: "*MNB*" indicate the F1-score of using Multinomial Naive Bayes, and "*SVM*" shows the results using Support Vector Machine.

of classification tries to classify two types of polarity of the subjective document. These polarities are positive or negative sentiment. We refer to it by (polarity_2). The last type is (polarity_3) which classifies three kinds of polarity. In this type, the neutral class is added to the classification process with the previous two polarities.

4.8.1. Baseline Experiment. In this experiment, we try to establish baseline results so that we can compare our next experiments according to that. As explained earlier, the baseline that we chose is the one that provides the basic knowledge about the text and might preserve the primary semantic feature of the language. Therefore, we use the Bag-Of-the-Word (BOW) feature model as a baseline model. We performed this experiment in different classification types and level and on different domains and ML classifiers.

Table 4.4 displays the results of the baseline experiment at the sentence level. The Table 4.5 illustrates the results at the document level classification. The baseline experiments are performed for subjectivity, polarity_2, and polarity_3, as explained earlier. These experiments are also carried out in five different domains. Table 4.5 shows the NA symbol in the case of subjectivity classification on movie reviews dataset. The NA means that the experiment is not applicable for this type of data because the movie review does not have any objective documents. Therefore, we cannot perform the subjectivity classification on that domain. This action has also been applied to the next experiments. The numbers in the tables refer to the weighted F1-score for each classifier that is calculated after performing 5-fold cross validation. The bold numbers show the best results for the classifiers in each domain for the baseline experiment. For example, the SVM achieves an F1 score of 85% in case of polarity_2 (positive versus negative) classification for the restaurant reviews field. In the same situation, the MNB achieves around F1 score of 84%.

From Tables 4.4 and 4.5, we notice that the SVM classifier outperforms the MNB in most cases. The MNB gets a good result in the case of polarity classification in one domain, the news. This suggests that the MNB would work better with the lengthy reviews that have more objective sentences, as what we have in newswire domains. Otherwise, we may infer that the SVM works better classifying most of the domains either the long or the short one. In addition, the SVM can find the sentiment better than the MNB even in the case of a lengthy review that has many objective sentences, such as movie review.

In regard to the domain of the dataset, we found that the newswire text is the hardest domain to classify, especially in subjectivity sentence level classification. This may indicate a difference between the domains. Some domains express the sentiment clearly, such as

the market reviews, whereas other domains show the subjective words and phrases in more complex manner, such as news text. In addition, the subjective words or phrase sometime could be included in other objective documents that makes this process more difficult than the polarity classification.

In the following experiments, we will use these results as a baseline to help us in judging the different feature sets. In addition, it gives evidence whether the classifier could learn new knowledge from the extra features. For the sake of readability, the results of the following experiments are rounded to integer numbers.

4.8.2. Different n-gram models. In this experiment, the different n-gram models were built and we investigate their effect on the different ML classifiers. The primary goal of this experiment is to find the n-gram model that works best with the sentence level classification in various types of sentiment classifications (subjectivity and polarity) for the Arabic text. The second goal is to figure out if the n-gram model could capture some of the relation between words and capture the effect of phrases on the sentiment, especially in the case of bi and tri-gram model.

To figure out these goals, we proposed and worked with only three different n-gram models. These models are uni-gram which is our baseline model, bi-gram, and tri-gram model. These n-gram models are the common ones that are used in sentiment analysis field (Pang and Lee, 2008). The main difference between these models are the number of words in each features rows of the model. For example, the feature row contains only one word in the case of the uni-gram model, whereas it has two words for the bi-gram model and so on. This difference affects the size of the features model that has been built. Table 4.6 shows the number of features that we have in the model for each different scenario. In addition, the

Table 4.6: Number of features in each model of different n-grams

	News Reviews	Restaurant Reviews	Market Reviews	Movie Reviews	Newswire
Uni-gram Baseline	29625	15737	3253	12499	90919
Uni+Bi-gram	112791	69080	10431	49301	12744
Uni+Tri-gram	118017	73478	10595	52423	49767
Uni+Bi+Tri-gram	201183	126821	17773	89225	53896

combination of these models also takes in our investigation. We did different mixing between these models and figured out the best combinations were using uni-gram alone, then adding bi-gram to it, and lastly adding tri-gram to them. The same approach is applied to both sentence and document level classification.

The results of comparing different n-gram models are depicted in Table 4.7 and 4.8. The initial experiments that have been done in the earlier preparation stages of this dissertation indicate using different n-gram such as bi- or tri-gram alone do not increase the accuracy of the classification. Therefore, we only consider adding the different n-grams to the uni-gram model which is the baseline model. The bold numbers show the best result achieved among all feature sets and classifiers. The underlined numbers indicate the best n-gram configuration setting that works better for the particular classifier.

Regarding the different n-gram configurations, the underlined results in Table 4.7 display that the uni-gram model which is our baseline model, achieves most of the best results in both classifiers. However, the different n-gram combinations, such as adding uni-gram with bi- or tri-gram, achieve similar results to the baseline. This could give us evidence that the classifier might or might not learn something from this feature. Therefore, using the uni-gram model for Arabic sentiment analysis is considered the best choice because the uni-gram model captures the basic unit of the sentences, which is a word. However, it is useful to use

Table 4.7: Different n-gram models at the sentence level classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Uni-gram Baseline	67%	**69%**	57%	**58%**	55%	**57%**
	Uni+Bi-gram	65%	**69%**	59%	55%	54%	**57%**
	Uni+Tri-gram	64%	**69%**	58%	56%	53%	**57%**
	Uni+Bi+Tri-gram	55%	68%	58%	54%	51%	**57%**
Restaurant Reviews	Uni-gram Baseline	70%	**71%**	81%	**83%**	72%	**73%**
	Uni+Bi-gram	69%	**71%**	81%	**83%**	72%	72%
	Uni+Tri-gram	69%	**71%**	81%	**83%**	71%	72%
	Uni+Bi+Tri-grams	68%	70%	81%	**83%**	70%	72%
Market Reviews	Uni-gram Baseline	88%	**89%**	87%	**88%**	69%	**69%**
	Uni+Bi-gram	85%	**89%**	87%	**88%**	70%	68%
	Uni+Tri-gram	86%	**89%**	87%	**88%**	69%	68%
	Uni+Bi+Tri-grams	81%	**89%**	88%	**88%**	69%	68%
Movie Reviews	Uni-gram Baseline	44%	45%	77%	**80%**	49%	**52%**
	Uni+Bi-gram	48%	38%	69%	**80%**	50%	51%
	Uni+Tri-gram	48%	38%	70%	**80%**	49%	51%
	Uni+Bi+Tri-grams	**49%**	36%	56%	**80%**	45%	49%
News	Uni-gram Baseline	33%	35%	**82%**	80%	**73%**	71%
	Uni+Bi-gram	37%	28%	79%	79%	67%	71%
	Uni+Tri-gram	**38%**	26%	79%	79%	68%	71%
	Uni+Bi+Tri-grams	37%	22%	77%	78%	54%	71%

bi- or tri-gram to capture some relationship between words, but this may add some noise to the data by adding unnecessary relationships with other words.

It is clear from Table 4.7 that the SVM achieves the best results for all classification types. In some cases, with some n-gram configurations, MNB behaves better than the SVM. For example, MNB increases the accuracy to 59% in the polarity_2 classification of news reviews domain. It seems that the SVM does not learn new knowledge from adding the extra n-gram to the feature model, whereas the MNB learns some new information about the sentiment problem in Arabic in some cases. However, adding a different n-gram to the model adds more noise the MNB classifier. Therefore, we could say that the SVM is not affected by the different n-gram models and is able to achieve the same accuracy of classification in

Table 4.8: Different n-gram models at the document level classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Uni-gram Baseline	86%	88%	54%	**56%**	**61%**	58%
	Uni+Bi-gram	68%	**89%**	56%	53%	56%	57%
	Uni+Tri-gram	64%	**89%**	56%	52%	55%	57%
	Uni+Bi+Tri-gram	37%	**89%**	53%	50%	44%	57%
Restaurant Reviews	Uni-gram Baseline	95%	**96%**	84%	**85%**	65%	**67%**
	Uni+Bi-gram	93%	**96%**	83%	**85%**	**67%**	66%
	Uni+Tri-gram	93%	**96%**	84%	**85%**	**67%**	66%
	Uni+Bi+Tri-grams	81%	**96%**	78%	**85%**	65%	64%
Market Reviews	Uni-gram Baseline	92%	93%	88%	**90%**	69%	**70%**
	Uni+Bi-gram	90%	94%	88%	89%	69%	68%
	Uni+Tri-gram	90%	**95%**	88%	89%	**70%**	**70%**
	Uni+Bi+Tri-grams	87%	**95%**	88%	89%	**70%**	69%
Movie Reviews	Uni-gram Baseline	NA	NA	78%	80%	**49%**	44%
	Uni+Bi-gram	NA	NA	26%	**81%**	48%	43%
	Uni+Tri-gram	NA	NA	10%	**81%**	39%	43%
	Uni+Bi+Tri-grams	NA	NA	0%	80%	17%	40%
News	Uni-gram Baseline	57%	63%	77%	76%	**69%**	65%
	Uni+Bi-gram	48%	65%	75%	78%	50%	64%
	Uni+Tri-gram	44%	**67%**	74%	**79%**	48%	64%
	Uni+Bi+Tri-grams	37%	**67%**	51%	**79%**	23%	60%

sentiment of Arabic language. However, this adds more size to the feature model. On the other hand, the MNB might be very sensitive about adding more n-gram models to the feature, so this may add more noise to it.

Table 4.8 presents the results of classification at the document level for the different n-grams models. It reflects the same observation of the sentence classification results with a slight difference. Regarding the various n-gram configurations, the uni-gram also seems to be the best n-gram model that might work with the ML classification in Arabic sentiment analysis. However, the other combination plays some roles with adding bi- or tri-gram with the uni-gram model. For example, the results of subjectivity classification in the newswire domain improves by 4% from 63% in the baseline to 67% when adding more n-grams in the

model. To find whether the difference between the results in Uni-gram and Uni_Tri-gram model is significant, we calculate the range of differences between the F_1 in each k-fold. In this same domain, news, the range in F1 with a uni-gram model was around ±8.8, whereas it was around ±5.3 with uni+tri-gram model in the subjectivity. In the case of polarity_2 with newswire domain, the difference was around ±6.5 using uni-gram model compared to ±2.3 using uni+bi+tri-gram model. This indicate that the uni+tri-gram model work more better in different k-fold experiments. Therefore, we suggest that the different n-gram models could be able to capture more relations between words that help in sentiment analysis for document level classification, which is not the case of the sentence level classification.

4.8.3. Morphological features. This section will focus on POS features and their effect in Arabic sentiment analysis. The first feature set is using the POS tag of the words, beside the word itself. The second feature is one that only relies on particular types of POS, the adjective and the adverb words. Some researchers (Benamara et al., 2007) found that using only adjectives and adverbs is enough to capture the sentiment in the text and reduce the space of feature model.

The primary purpose of this experiment is to find the effect of adding some morphological knowledge to the feature model. In addition, this analysis tries to find the most effective morphological feature. Finally, it helps to investigate the impact of using these features for the first time on the Dialect language type of Arabic.

Table 4.9 and Table 4.10 show the results of the MNB and the SVM classifiers with two feature models that include morphology characteristics of the Arabic words in both classification levels. Table 4.9 illustrates results for the sentence level and Table 4.10 in is for the document level. The bold number depicts the best outcome achieved among the

Table 4.9: Basic morphology tag as a feature at the sentence level classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Baseline	67%	69%	57%	58%	55%	57%
	With POS	72%	69%	60%	56%	57%	55%
	With Adv_Adj	**73%**	70%	**61%**	58%	**58%**	57%
Restaurant Reviews	Baseline	70%	71%	81%	83%	72%	**73%**
	With POS	73%	70%	83%	81%	71%	72%
	With Adv_Adj	**74%**	71%	**84%**	83%	72%	**73%**
Market Reviews	Baseline	88%	89%	87%	88%	69%	69%
	With POS	**90%**	**90%**	**91%**	89%	**70%**	69%
	With Adv_Adj	**90%**	88%	**91%**	89%	**70%**	69%
Movie Reviews	Baseline	44%	**45%**	77%	80%	49%	52%
	With POS	44%	**45%**	77%	81%	51%	56%
	With Adv_Adj	44%	44%	77%	**82%**	51%	**57%**
News	Baseline	33%	35%	**82%**	80%	**73%**	71%
	With POS	**37%**	35%	**82%**	80%	**73%**	71%
	With Adv_Adj	**37%**	**37%**	**82%**	80%	**73%**	71%

Key: "*With POS*" indicate the F1-score of using POS feature with the baseline model, and "*With Adv_Adj*" shows the results using the Adv_Adj feature with the baseline model.

Table 4.10: Basic morphology tag as a feature at the document level classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Baseline	86%	**88%**	54%	56%	**61%**	58%
	With POS	85%	**88%**	62%	56%	60%	58%
	With Adv_Adj	86%	**88%**	**63%**	55%	**61%**	58%
Restaurant Reviews	Baseline	95%	**96%**	84%	85%	65%	**67%**
	With POS	**96%**	**96%**	86%	85%	66%	66%
	With Adv_Adj	95%	**96%**	**87%**	86%	65%	**67%**
Market Reviews	Baseline	92%	93%	88%	90%	69%	70%
	With POS	**94%**	**94%**	92%	90%	69%	70%
	With Adv_Adj	**94%**	93%	**93%**	91%	70%	**71%**
Movie Reviews	Baseline	NA	NA	78%	80%	49%	44%
	With POS	NA	NA	79%	81%	**51%**	**51%**
	With Adv_Adj	NA	NA	76%	**82%**	46%	48%
News	Baseline	57%	63%	77%	76%	69%	65%
	With POS	61%	**65%**	**81%**	76%	69%	66%
	With Adv_Adj	57%	63%	**81%**	77%	**70%**	63%

Key: "*With POS*" indicate the F1-score of using POS feature with the baseline model, and "*With Adv_Adj*" shows the results using the Adv_Adj feature with the baseline model.

classifiers for each classification type. For example, the SVM achieves an F1 of 82% to classify polarity_2 type at document level in movie reviews. The underlined numbers display the best results that are achieved using a particular feature model for each classifier and classification type. For example, both classifiers achieve the best F1 score using (Adv_Adj) feature model in subjectivity classification at the sentence level in news reviews dataset.

It can be noticed that adding some Arabic morphological features improved classification for some datasets. In addition, these features do not affect the performance of the classifier compared to the baseline model. For example, the morphology features help to improve the results of both classifiers for polarity_2 classifier at document level in restaurant reviews. This improvement might be coming from the knowledge that has been added by the POS feature model. Morphological features add new characteristics to the model that can distinguish between the words that have the same letters but have a different meaning. In Arabic language, many words might have the same structure but they might be used in different meaning. For example, the word (**سعيد**/ *sʕyd* / Saed or happy) may refer to the male person proper noun "Saed" or may reflect an adjective meaning "happy". Determining the difference between these meanings without knowing its POS is difficult for the classifier. Therefore, the Morphological tag, that is POS, adds this knowledge to the classifier.

Some of the varying results in different domains may be due to the use of the Arabic morphology tools library (AMIRA) (Diab, 2009). This tool was built and trained on one type of Arabic, MSA. The POS that came from the AMIRA tool may not reflect the actual POS of words in dialect Arabic in some cases. Therefore, this may affect the use of the morphology feature on some domains of the dataset or add some noise to the classifier. In this case, either a morphology tool that works well with Dialect Arabic is required in order

to use its outcome as a feature, or it will be necessary to rely on another method that might capture some of the morphology features of the Dialect language. For example, the results were decreased by 1% in case of Polarity_3 classification at the sentence level in the restaurant review. This issue might come from the morphological analyzer that has been used because it is not built for Dialect Arabic which is the nature of the dataset.

4.8.4. Stylistic features. Some of the research claims that stylistic features plays some roles in sentiment analysis (Abbasi et al., 2008; Pang and Lee, 2008). For example, they show that short sentences might be positive in most cases, whereas the long ones might be negative. The goal of this experiment is to investigate this concept in the Arabic text for both types of languages (MSA and DA). Finally, it is possible to figure out if this plays an important role in case of Arabic or not. In addition, this goal can be applicable to the document level classification as well as to the sentence level.

To do this experiment, we add only one feature. At the sentence level classification, the best feature that might be used is the number of words in the sentence. This feature shows the length of the sentence. We add this feature to our baseline. That means that the model will contain the BOW of the uni-gram model and the last column will show the number of the word in each sentence. In the document level classification, the extra column in the features will refer to the number of sentences in the documents. This information does not need to be calculated because it was already generated while building the corpus. Chapter 3 has more information about that. The first line of the document has the number of sentences in that document. We need only read this information and put it in the feature model.

Table 4.11: Number of words with the baseline model as a feature at the sentence level Classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Baseline	67%	69%	57%	**58%**	55%	**57%**
	With No. of words	**72%**	71%	52%	57%	53%	56%
Restaurant Reviews	Baseline	70%	71%	81%	**83%**	72%	**73%**
	With No. of words	**74%**	72%	**83%**	82%	68%	**73%**
Market Reviews	Baseline	88%	89%	87%	88%	69%	69%
	With No. of words	**91%**	86%	**89%**	88%	67%	**70%**
Movie Reviews	Baseline	44%	**45%**	77%	**80%**	49%	**52%**
	With No. of words	33%	44%	79%	79%	45%	51%
News	Baseline	33%	35%	82%	80%	73%	71%
	With No. of words	30%	**37%**	**83%**	79%	**74%**	70%

Key: "*With No. of words*" indicate the F1-score of using number of words feature with the baseline model.

Table 4.12: Number of sentence with the baseline model as a feature at the document classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Baseline	86%	**88%**	54%	56%	**61%**	58%
	With No. of sentences	**88%**	**88%**	**61%**	57%	60%	58%
Restaurant Reviews	Baseline	95%	**96%**	84%	85%	65%	**67%**
	With No. of sentences	**96%**	**96%**	**86%**	85%	62%	**67%**
Market Reviews	Baseline	92%	93%	88%	**90%**	69%	**70%**
	With No. of sentences	**94%**	93%	**90%**	89%	67%	**70%**
Movie Reviews	Baseline	NA	NA	78%	**80%**	**49%**	44%
	With No. of sentences	NA	NA	79%	**80%**	47%	43%
News	Baseline	57%	63%	77%	76%	69%	65%
	With No. of sentences	58%	**65%**	**81%**	77%	**71%**	66%

Key: "*With No. of* sentence" indicate the F1-score of using number of words feature with the baseline model.

Table 4.11 and Table 4.12 illustrate the results of using stylistic feature in Arabic sentiment analysis. The results of the sentence level classification are shown in Table 4.11 and document level in Table 4.12.

In the document classification level, we notice this feature tends to improve the performance of both classifiers in the news domain at all classification types. For example in Table

4.12 , it improved the result by more than 6% using MNB in the case of the news reviews domain, compared with the baseline, which used the uni-gram model at the sentence level in subjectivity classification. In addition, we calculate the range of difference between the F_1 in each k-fold. In this same domain, news reviews, the range in F_1 with baseline model was around ± 4.1, whereas it was around ± 2.8 with number of sentence model in the polarity_2. This also works in most cases of the other dataset domains except the movie reviews. This issue might come from the nature of the domain itself. The movie review usually has more sentences describing the plots of the film. In other domains, this behavior is not there because the user tends to include sentiment and factual information in each sentence. The movie and news domains contain more factual information, which may have added some noise to this feature, making it not too useful in the case of polarity using this feature. In the restaurant reviews domain, the results of subjectivity and polarity was improved. For example, the result of MNB increases to 74% with subjectivity which was 70% in the baseline model. The range of difference the F_1 in each k-fold was ± 2.8 in both models.

Similar to what found when adding the morphological feature, this feature, in general, adds more knowledge to the MNB compared to the SVM. However, this feature dose improve the SVM results in some cases. It improves the subjectivity classification at the sentence level in the news domain.

This feature achieved higher performance in the classifier in most cases with the subjectivity classification. This feature do not need more time to be computes. It also dose not hurt the performance of the classifier significantly. Therefore, we prefer to add and use this feature in Arabic sentiment analysis because it might add valuable knowledge to the classifier.

Table 4.13: Comparing linear and non-linear the SVM at the sentence level classification

		Subjectivity		Polarity_2		Polarity_3	
		LSVM	NLSVM	LSVM	NLSVM	LSVM	NLSVM
News	Uni-gram Baseline	69%	**70%**	**58%**	57%	57%	57%
Reviews	Uni+Bi-gram	69%	**70%**	55%	55%	57%	57%
	With POS	69%	69%	56%	**57%**	55%	**56%**
	With No. of words	71%	71%	57%	57%	56%	**57%**
Restaurant	Uni-gram Baseline	71%	**72%**	**83%**	82%	73%	73%
Reviews	Uni+Bi-gram	71%	71%	83%	83%	72%	72%
	With POS	70%	70%	81%	**82%**	72%	72%
	With No. of words	72%	72%	82%	82%	73%	73%
Market	Uni-gram Baseline	89%	89%	**88%**	87%	69%	69%
Reviews	Uni+Bi-gram	89%	89%	88%	88%	68%	**69%**
	With POS	90%	90%	89%	89%	69%	69%
	With No. of words	86%	86%	88%	**89%**	70%	70%
Movie	Uni-gram Baseline	45%	45%	**80%**	79%	52%	52%
Reviews	Uni+Bi-gram	38%	38%	80%	80%	51%	51%
	With POS	45%	**46%**	81%	81%	56%	56%
	With No. of words	44%	**45%**	79%	79%	51%	51%
News	Uni-gram Baseline	35%	**36%**	80%	80%	**71%**	70%
	Uni+Bi-gram	28%	28%	79%	79%	71%	71%
	With POS	35%	**36%**	80%	80%	71%	71%
	With No. of words	37%	37%	**79%**	78%	70%	70%

4.8.5. Comparing Linear and Non_Linear kernel of the SVM. This section will evaluate the performance of using a non-linear kernel for the SVM and compare it with the linear kernel of the SVM. This will indicate whether Arabic sentiment analysis needs more complicated representation or not. In order to investigate this idea, some of features were chosen to perform both linear and nonlinear kernels of the SVM. These features are the base models, which are Uni-gram model, Uni-gram with bi-gram, Uni-gram with POS, and uni-gram with stylistic features (number of sentences or words). The polynomial function is used with 2 degree to perform the nonlinear kernel of the SVM.

Table 4.13 and Table 4.14 show the detailed results of using Linear and Non-Linear kernel the SVM for sentence and document level classification. The LSVM refers to the

Table 4.14: Comparing linear and non-linear the SVM at the document level classification

		Subjectivity		Polarity_2		Polarity_3	
		LSVM	NLSVM	LSVM	NLSVM	LSVM	NLSVM
News	Uni-gram Baseline	88%	88%	56%	56%	58%	58%
Reviews	Uni+Bi-gram	89%	89%	53%	53%	57%	**58%**
	With POS	88%	88%	56%	56%	58%	58%
	With No. of sentences	88%	88%	57%	57%	58%	58%
Restaurant	Uni-gram Baseline	96%	96%	85%	85%	67%	67%
Reviews	Uni+Bi-gram	96%	96%	85%	85%	66%	66%
	With POS	96%	96%	85%	85%	**66%**	64%
	With No. of sentences	96%	96%	85%	85%	67%	67%
Market	Uni-gram Baseline	93%	93%	**90%**	89%	70%	70%
Reviews	Uni+Bi-gram	94%	94%	89%	89%	68%	68%
	With POS	94%	**95%**	**90%**	89%	**70%**	69%
	With No. of sentences	93%	93%	89%	**90%**	70%	70%
Movie	Uni-gram Baseline	NA	NA	80%	80%	**44%**	43%
Reviews	Uni+Bi-gram	NA	NA	81%	81%	43%	43%
	With POS	NA	NA	81%	81%	51%	51%
	With No. of sentences	NA	NA	80%	80%	43%	43%
News	Uni-gram Baseline	63%	63%	76%	76%	65%	65%
	Uni+Bi-gram	65%	65%	**78%**	77%	64%	64%
	With POS	65%	65%	76%	76%	66%	66%
	With No. of sentences	65%	65%	77%	77%	66%	66%

linear kernel of the SVM classifier. The NLSVM displays the non-linear kernel of the SVM results. For each feature sit the best result is boldfaced. In both tables, we have noticed that both classifiers achieve a similar performance. For example, the LSVM achieves the best result in the case of subjectivity classification in the market review domain with three different feature sets. The NLSVM achieves the best performance using uni- and bi-gram models together. The difference between the performances of the classifiers was around 1%, which is insignificant.

In the sentence level subjectivity classification, the NLSVM achieved the best results with two different domains compared to the LSVM. In the case of polarity_2 classification, the LSVM outperformed the other classifier by achieving the best results in one domain

compared to the NLSVM. The situation was changed in the case of polarity_3 classification; the LSVM and the NLSVM achieved the similar performance. It seems that the NLSVM dominated the LSVM classifier in the case of sentence level classification. However, there were comparable results between the LSVM and NLSVM. Since the difference between the results achieved by the LSVM and the NLSVM was less than or equal 1%, we conclude that the linear SVM is adequate and we only use linear kernel in the remaining of our experiment.

According to the previous preservation, it may reveal the fact that other classifiers, such as non-linear ones, may work better in discovering the sentiment in a long Arabic text. Therefore, more investigation is necessary in this area to either prove or disprove the fact that long Arabic text requires more non-linear classifiers to establish the sentiment rather than using the linear ones. The neural networks is another nonlinear popular classifier and is investigated in Chapter 6.

4.8.6. Advanced Features. The following sections show the details of our experiments using the proposed Advanced feature sets that might be used to build feature models. These features might be applied to either document of sentence level. One of them only is used in the document level classification because it mainly captures the position of the opinioned sentence in the document. In each section, the description of the feature will be provided and followed by an explanation of how they are applied to Arabic sentiment analysis. The results and findings will also be discussed at the end of each section.

4.8.7. Position of Opinioned Sentence on a Document. As explained earlier, this feature represents the position of the sentence in the text. In this experiment, the investigation of the effect of using the proposed feature is carried out in order to find if the classifier performance is better if we only consider sentences at the beginning and ending of

Table 4.15: Result of comparing baseline with without using position approach

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
Restaurant Reviews	Baseline	95%	**96%**	84%	**85%**	65%	**67%**
	With Position	95%	**96%**	82%	84%	63%	65%
Movie Reviews	Baseline	NA	NA	78%	80%	49%	44%
	With Position	NA	NA	69%	**83%**	**50%**	41%
News	Baseline	57%	63%	77%	76%	**69%**	65%
	With Position	55%	**67%**	75%	**78%**	61%	62%

Key: "*With P*osition" indicate the F1-score of using sentence position feature with the baseline model.

a document. In addition, it will contribute to prove whether the primary sentiment of the text is preserved in the proposed position of the sentence in Arabic language.

In order to perform this experiment, only long documents were included to this part. In our corpus, only two domains are considered as long documents, which are the movie reviews and the news domains. Chapter 3 shows the average number of sentences in each document for each dataset domain. The common locations of the text that might carry the main opinion of the document are at the beginning and ending of the document. Therefore, these positions are used to build the feature model. To evaluate this proposed model, we will compare the performance of the classifier with the baseline that we have. The baseline contains all words of the document, whereas the proposed one should only have part of those words which are locations in either the beginning or ending of the document. If the performance of the proposed method outperforms the baseline, then the words that do not contribute to and may deviate from sentiment analysis.

The performance of the classifier using the sentence position method is shown in Table 4.15. In this table, the bold numbers illustrate the best result that is achieved either using baseline with or without the position approach in various classifiers. For example, the SVM achieves the best result with 78% using polarity_2 classification in the newswire domain.

The underlined numbers represent the best approach that works for each classifier. For example, the performance of the SVM increased by 3% using position approach in polarity_2 classification in the Movie Reviews domain.

It is clear that using the position of the sentence approach works in some cases from the data shown in Table 4.15. With polarity_2 classification, we noticed that this approach plays a main role in increasing the accuracy of the SVM. This method also helps in the case of news domain by increasing the result by 4%. On the other hand, the proposed method would not work well if used in the restaurant reviews field. This issue might come from the nature of the domain. Both domains, news and movie, are considered as long texts, whereas the restaurant is medium to small text. Therefore, the text of the restaurant domain may not reflect the sentiment orientation in the different positions of the text. This may affect the performance of using our proposed position approach with this type of domain.

4.8.8. Base Phrase Chunk. This experiment aims to perform and evaluate the effect of the proposed feature (BPC) on Arabic sentiment analysis. This is done on document and sentence classification levels. In addition, it tries to indicate whether there is a relation between the BPC and sentiment in Arabic text. This feature might capture the context meaning of the text because it preserves the relations between words. In sentiment, some words can carry the particular sentiment orientation by itself but they could convey a different sentiment when they are grouped with other words in one phrase.

Tables 4.16 and 4.17 show the result of using BPC feature with a different classifiers and domains. They also compare the use of the BPC feature with the baseline feature model alone. The bold numbers illustrate the best result among different classification processes and the underlined numbers show the feature model that work better with different classification

Table 4.16: Advanced morphology tag BPC as a feature at the sentence level Classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Baseline	67%	69%	57%	58%	55%	57%
	With BPC	**72%**	69%	**61%**	57%	**58%**	56%
Restaurant Reviews	Baseline	70%	71%	81%	83%	72%	**73%**
	With BPC	**74%**	70%	**84%**	83%	72%	**73%**
Market Reviews	Baseline	88%	**89%**	87%	88%	69%	69%
	With BPC	88%	**89%**	**91%**	90%	**71%**	69%
Movie Reviews	Baseline	44%	45%	77%	80%	49%	52%
	With BPC	**48%**	44%	75%	**82%**	52%	**56%**
News	Baseline	33%	35%	**82%**	80%	**73%**	71%
	With BPC	**36%**	34%	**82%**	80%	72%	72%

Key: "*With BPC*" indicate the F1-score of using Based Phrase Chunk feature with the baseline model.

type. For example, the best result is achieved by using BPC approach and MNB with subjectivity classification in the movie reviews domain.

In Table 4.16, the result of polarity_3 classification of movie reviews improves by 4%. In addition, the range of difference between the F_1 was ± 3.5 in baseline and ± 2.1 in the BPC model. The performance of the SVM increases by 8% in the case of document classification with the same domain. The range of difference between the F_1 was around ± 6 in both model.

Most of the time, adding BPC to the Baseline model improves the results. The knowledge of the BPC method includes a basic syntax or structure of the sentence. That structure might preserve the actual sentiment orientation that is in the sentence or the text. This knowledge is useful sometimes to know which phrase contains the word that belongs to it. As a result, the actual sentimental word meaning will be preserved along with that phrase.

The BPC feature model approach sometimes hurts the performance of the sentiment classification process. These negative results are seen in the market reviews, news reviews, and restaurant reviews domains. AMIRA (Diab, 2009) is trained with MSA. The authors' of the AMIRA tool claimed that their tools might also be used with DA. Therefore, we used

Table 4.17: Advanced morphology tag BPC as a feature at the document level classification

		Subjectivity		Polarity_2		Polarity_3	
		MNB	SVM	MNB	SVM	MNB	SVM
News Reviews	Baseline	86%	88%	54%	56%	**61%**	58%
	With BPC	85%	**89%**	**62%**	57%	**61%**	58%
Restaurant Reviews	Baseline	95%	**96%**	84%	85%	65%	**67%**
	With BPC	95%	**96%**	**87%**	86%	66%	**67%**
Market Reviews	Baseline	92%	**93%**	88%	90%	69%	70%
	With BPC	**93%**	**93%**	**93%**	90%	70%	**71%**
Movie Reviews	Baseline	NA	NA	78%	80%	49%	44%
	With BPC	NA	NA	76%	**81%**	46%	**52%**
News	Baseline	57%	63%	77%	76%	**69%**	65%
	With BPC	60%	**64%**	**82%**	78%	67%	64%

Key: "*With BPC*" indicate the F1-score of using Based Phrase Chunk feature with the baseline model.

it to get the BPC tag of the Arabic text. The three domains, Market reviews, restaurant reviews, and news reviews, contain DA in most parts of their data. Therefore, the AMIRA may work well in some locations of this data, whereas it would not work with some other parts of the Dialect Arabic text. As a result of that, the use AMIRA in BPC might hurt the performance of the classification process with this type of language.

4.8.9. Comparing BPC with POS. By looking at Table 4.16, 4.17, 4.9, and 4.10 compare the results of using BPC features with other morphology features. From these tables, we can notice than the BPC sometimes works better than POS and the Adv_Adj features. The BPC also reduces the performance by 1% to 2% in some cases compared to the POS and Adv_Adj feature. However, the BPC was able to increase the performance by 4% where the other morphology features were not. For example, the performance of the classifier improved by 4% in the case of subjectivity classification of sentence level in the movie reviews domain. This may reflect that the BPC has some unique information that is added to the classifier knowledge. As explained earlier, the BPC shows the basic syntax structure of the text that is not there in the case of other morphology features such

as POS. Therefore, using BPC feature helps to improve the classifier processing which does not happen with other morphology features.

4.8.10. Polarity Feature. One ML classifier will be used to assess the proposed advanced feature setting. The SVM with linear kernel was chosen because it is considered to be the state-of-the-art classifier that has been used a lot in sentiment analysis problem. The primary goal is to evaluate the effect of using the concept of polarity score as a feature in the feature model and to see if that adds some useful knowledge to the classifier.

On of the goal of this approach is to fill the gap of missing a suitable primary NLP tool for dialect Arabic that is used to get morphological features of the text. As explained earlier, building morphological analyzer for one type of Arabic language dose not necessary work for another kind. In addition, there are different types of Dialect Arabic which leads to building a particular morphological analyzer for each of them. This is time consuming and costly. Not only that, there is a lack of specialized Dialect corpora that might help to build those tools. However, the different type of Arabic languages share most of the origin words with some variations in term of words of syntax. Therefore, using the polarity score of the word might be a reasonable approach.

Another objective of this experiment relates to evaluating the effect of the translation machine mechanism on sentiment analysis. The way of calculation a polarity score method depends on the machine translation concept. By performing this experiment, we may infer that the translation machine mechanism would preserve the sentiment orientation of the source language.

Table 4.18: Adding polarity score and count as a feature at the sentence level Classification

		Subjectivity		Polarity_2		Polarity_3	
		Count	Score	Count	Score	Count	Score
News Reviews	Baseline	69.2%	69.2%	58.1%	58.1%	57.3%	57.3%
	Pol No Stem	70.1%	70.1%	**58.3%**	58.0%	57.4%	57.2%
	Pol with Stem	**70.4%**	70.1%	58.0%	**58.3%**	**57.6%**	57.4%
	Pol Stem Only	70.2%	**70.5%**	58.2%	58.2%	57.4%	**57.6%**
Restaurant Reviews	Baseline	71.0%	71.0%	83.4%	**83.4%**	73.2%	73.2%
	Pol No Stem	71.4%	71.3%	83.5%	83.3%	**73.3%**	**73.3%**
	Pol with Stem	**71.5%**	**71.4%**	**83.9%**	83.3%	73.2%	73.2%
	Pol Stem Only	71.3%	**71.4%**	83.3%	**83.4%**	73.2%	**73.3%**
Market Reviews	Baseline	**89.3%**	**89.3%**	88.2%	88.2%	69.4%	69.4%
	Pol No Stem	89.0%	**89.3%**	**88.3%**	88.3%	69.4%	69.3%
	Pol with Stem	**89.3%**	**89.3%**	88.1%	88.1%	**69.6%**	68.9%
	Pol Stem Only	89.1%	88.7%	88.0%	**88.4%**	69.2%	**69.5%**
Movie Reviews	Baseline	**45.0%**	**45.0%**	80.0%	80.0%	52.1%	52.1%
	Pol No Stem	44.7%	44.9%	**80.4%**	**80.8%**	52.3%	52.7%
	Pol with Stem	44.6%	44.8%	79.4%	80.1%	52.3%	**52.8%**
	Pol Stem Only	44.9%	44.9%	80.3%	80.0%	**52.4%**	52.4%
News	Baseline	35.2%	35.2%	80.1%	80.1%	71.2%	**71.2%**
	Pol No Stem	35.8%	35.6%	81.3%	81.0%	**71.6%**	70.8%
	Pol with Stem	35.7%	**36.3%**	**81.5%**	80.7%	71.3%	70.6%
	Pol Stem Only	**36.0%**	36.0%	80.4%	**81.3%**	70.5%	70.9%

Key: "*Pol No S*tem" indicate the F1-score of using Polarity feature without using stem method with the baseline model, and "*Pol with S*tem" shows the results using Polarity feature with using stem method when the actual word does not have translation, and "*Pol Stem Only*" displays the results using Polarity feature using the stem on the word firstly before the translation.

Table 4.18 shows the results of using polarity approach at the sentence level classification. The document level classification results of this approach are displayed in Table 4.19. Different configurations and combinations are used while injecting the polarity concept with the feature model. The first row in these tables represents the results of baseline model feature, that is BOW. The next three rows show a different configuration using polarity concept with feature model. "PolNoStem" refers to the method when the polarity is computed without using stem technique as explained earlier. The "PolWithStem" represents using stem when the word dose not translate. "PolStemOnly" represents results when the stem is applied first on the word before the translation. For each type of classification and different polarity

Table 4.19: Adding polarity score and count as a feature at the document Level Classification

		Subjectivity		Polarity_2		Polarity_3	
		Count	Score	Count	Score	Count	Score
News Reviews	Baseline	88.1%	88.1%	56.4%	56.4%	58.1%	58.1%
	Pol No Stem	88.1%	88.1%	55.9%	56.4%	58.1%	**58.4%**
	Pol with Stem	88.1%	88.1%	56.4%	55.8%	57.9%	57.1%
	Pol Stem Only	**88.2%**	**88.2%**	55.9%	**56.5%**	57.4%	58.1%
Restaurant Reviews	Baseline	**96.2%**	**96.2%**	**85.3%**	**85.3%**	67.0%	67.0%
	Pol No Stem	95.9%	95.9%	84.5%	84.6%	66.8%	66.9%
	Pol with Stem	95.9%	95.9%	84.4%	84.8%	**67.5%**	67.0%
	Pol Stem Only	95.9%	95.9%	**85.3%**	84.7%	66.9%	**67.3%**
Market Reviews	Baseline	**93.4%**	**93.4%**	90.0%	90.0%	70.0%	70.0%
	Pol No Stem	**93.4%**	**93.4%**	90.3%	**90.5%**	**70.1%**	**70.3%**
	Pol with Stem	93.1%	93.1%	**90.5%**	89.9%	**70.1%**	70.2%
	Pol Stem Only	93.2%	93.2%	89.9%	**90.5%**	69.7%	69.9%
Movie Reviews	Baseline	NA	NA	**80.0%**	80.0%	44.5%	44.5%
	Pol No Stem	NA	NA	78.1%	79.2%	**46.5%**	**46.3%**
	Pol with Stem	NA	NA	79.0%	**80.1%**	44.4%	43.3%
	Pol Stem Only	NA	NA	78.1%	79.2%	**46.5%**	45.8%
News	Baseline	63.4%	63.4%	76.4%	**76.4%**	65.3%	**65.3%**
	Pol No Stem	64.3%	63.6%	75.6%	**76.4%**	65.4%	64.6%
	Pol with Stem	**64.8%**	62.5%	**76.9%**	75.7%	**65.8%**	64.2%
	Pol Stem Only	63.4%	**64.2%**	74.5%	75.2%	63.4%	63.7%

Key: "*Pol No S*tem" indicate the F1-score of using Polarity feature without using stem method with the baseline model, and "*Pol with S*tem" shows the results using Polarity feature with using stem method when the actual word does not have translation, and "*Pol Stem Only*" displays the results using Polarity feature using the stem on the word firstly before the translation.

configuration, two mechanisms are used to add polarity into the feature model. The first column refers to the counting method, when the polar words are counted. The second for the score method when the total score is calculated for each type of polarity.

Most of the time, using the polarity either outperforms or is similar to the baseline model. The polarity approach helps more in the case of polarity classification than the subjectivity because it may add more detail about the polarity aspect than the subjectivity orientation. However, this method adds more performance to the classifier in the case of subjectivity. For example, the result increased by more than 1 % in the event of subjectivity for the news

domain. This trend of increasing the performance can be found in all types and levels of sentiment classification of Arabic language.

Regarding the best mechanisms that should be used with polarity method, we notice that using stem mechanism with polarity approach helps to improve the performance of the proposed model, especially in the dataset domain that has Dialect Arabic language. This might come from the nature of the data itself. We have different Dialect words that might come from the same MSA Arabic but is not found in MSA as an actual word. The translation engine only works well with the MSA Arabic. Therefore, using the stem in some cases may help the translation to find appropriate work in both types of Arabic language, MSA and DA. For example, the result increases by 0.9% with polarity_2 classification in restaurant review domain, Table 4.18.

In order to make a judgment on the best polarity techniques (counting or scoring) that should be used, we calculate which method achieves the best result in each classification process. Table 4.20 illustrates this comparison. For example, the Polarity Counting method achieves the best result three times compared to nine times for Polarity Scoring method using "PolNoStem" feature model in all classification types in the document classification level. We have noticed that the scoring technique is outperforming the counting in the case of document level classification with 20 times versus 13 times. On the other hand, the counting achieves 19 best results versus 14 in the case of the sentence level classification. This suggests that counting polar words is better than calculating their score in case of the sentence level. That means the score value of the total polarity only works best for the long text and the counting method works best for the short text such as the sentence. The other observation that we can infer from the data in Table 4.20 is the counting technique works better with

Table 4.20: Counting versus Scoring in each Document and Sentence Classification

Model	Document		Sentence	
	Counting	Scoring	Counting	Scoring
Pol No Stem	3	9	9	2
Pol with Stem	9	2	9	4
PolStemOnly	2	9	1	9

Key: "*Pol No S*tem" indicate the F1-score of using Polarity feature without using stem method with the baseline model, and "*Pol with S*tem" shows the results using Polarity feature with using stem method when the actual word does not have translation, and "*Pol Stem Only*" displays the results using Polarity feature using the stem on the word firstly before the translation. "Counting" indicates the method of counting the polar words, whereas the "Scoring" shows the method of using the calculating the polarity score.

the "PolWithStem" model than the Scoring. The Scoring works well with applying stem first in "PolStemOnly" model. This might help if we want to reduce the effect of the actual word and use the stem technique.

4.8.11. Word Clustering. The goal of this experiment is to evaluate whether using the word cluster tag adds some sentiment knowledge to the classifier. It also tries to check whether the same cluster group has the same sentiment words.

Table 4.21 displays the experiment of using a cluster method during Arabic sentiment analysis at sentence level classification. This method uses the cluster ID of the words as a feature to build feature model. We compare using different cluster groups to find the best cluster group that might work well with sentiment analysis. The BOW baseline model was used to evaluate the performance of the cluster approach. The bold numbers illustrate the best results that are recorded using a particular feature model setting. Only the SVM classifier is used to evaluate the effect of the cluster idea.

Using cluster ID of the words as a feature is not useful in most cases. It is clear that the BOW baseline feature has the best performance results compared to all the different cluster configurations. For example, the best results were achieved using BOW model in

subjectivity classification for all dataset domains. The F1 score decreased by more than 10% when the cluster ID is used to build feature model. However, there are some positive sides to using the cluster method that inspired us to make some enhancements to this method.

Notices by results at the document level in Table 4.22, the F1 score was improved by 2% in the case of polarity_3 classification in the movie reviews domain. In news domain, the result was increased in polarity_2 classification by 3%. This might infer that the cluster might play some role in the polarity classification process and might preserve some of the sentimental orientation across different cluster groups. The other improvement was noticed with increasing F1 scores with increasing the cluster groups. The F1 score of the 50 cluster group is very small comparing to the BOW. However, the F1 score was improved by adding more cluster groups. For example, the cluster of 50 groups achieves 77% F1 score then it increases until it reaches to 93% with using 1000 clusters during subjectivity classification process in the restaurant domain. These two improvements inspire us to do more investigating using the cluster method.

Table 4.21 shows the results using the same approach that we applied with the results in Table 4.22 but in the Sentence-Level classification. The results illustrate similar findings that we recorded in the document level classification. The performance was not improved in most cases except that it improves by 2% with subjectivity classification in movie reviews and news domains. The trend of improvement with different cluster groups is the same that we noticed in document level classification. Therefore, the next experiment investigates using a cluster method in a different manner.

Tables 4.23 and 4.24 show the results of the sentence and document level classification process using a cluster approach enhancement. For the previous experiment, we noticed that

Table 4.21: Cluster ID as name in feature model at the sentence level classification

		Subjectivity	Polarity_2	Polarity_3
		SVM	SVM	SVM
News Reviews	BOW	**69%**	**58%**	**57%**
	Cluster 50	44%	50%	36%
	Cluster 100	50%	49%	39%
	Cluster 500	57%	53%	48%
	Cluster 1000	62%	52%	48%
Restaurant Reviews	BOW	**71%**	**83%**	**73%**
	Cluster 50	55%	55%	47%
	Cluster 100	61%	60%	49%
	Cluster 500	69%	71%	56%
	Cluster 1000	70%	71%	59%
Market Reviews	BOW	**89%**	**88%**	**69%**
	Cluster 50	80%	77%	56%
	Cluster 100	78%	80%	58%
	Cluster 500	84%	84%	65%
	Cluster 1000	85%	86%	67%
Movie Reviews	BOW	45%	**80%**	**52%**
	Cluster 50	41%	70%	46%
	Cluster 100	46%	78%	46%
	Cluster 500	**47%**	73%	44%
	Cluster 1000	46%	71%	47%
News	BOW	35%	**80%**	**71%**
	Cluster 50	36%	61%	48%
	Cluster 100	36%	63%	54%
	Cluster 500	35%	69%	60%
	Cluster 1000	**37%**	69%	61%

1000 cluster groups achieved the best result compared to the other clusters. Therefore, we only considered this cluster in our enhancement process. We then merge the cluster ID of the word with the word itself as the method of POS feature. We then compare using this feature with the BOW baseline model. Most of the time, the new enhanced cluster features, which is shown in the second row for each dataset domain, achieve the best performance. For example, the F1 score increases by 3% in polarity classification in market review domain, Table 4.23.

Table 4.22: Cluster ID as name in feature model at the document Level classification

		Subjectivity	Polarity_2	Polarity_3
		SVM	SVM	SVM
News Reviews	BOW	**88%**	**56%**	**58%**
	Cluster 50	75%	51%	43%
	Cluster 100	74%	50%	46%
	Cluster 500	79%	55%	53%
	Cluster 1000	81%	54%	52%
Restaurant Reviews	BOW	**96%**	**85%**	**67%**
	Cluster 50	77%	69%	50%
	Cluster 100	79%	73%	50%
	Cluster 500	91%	74%	53%
	Cluster 1000	93%	77%	58%
Market Reviews	BOW	**93%**	**90%**	**70%**
	Cluster 50	83%	80%	59%
	Cluster 100	79%	81%	59%
	Cluster 500	89%	87%	66%
	Cluster 1000	90%	87%	68%
Movie Reviews	BOW	NA	**80%**	44%
	Cluster 50	NA	76%	44%
	Cluster 100	NA	71%	44%
	Cluster 500	NA	74%	44%
	Cluster 1000	NA	75%	**46%**
News	BOW	**63%**	76%	**65%**
	Cluster 50	45%	71%	53%
	Cluster 100	47%	77%	63%
	Cluster 500	59%	70%	61%
	Cluster 1000	58%	**79%**	64%

Table 4.23: Combine baseline model with cluster feature model at the sentence level classification

		Subjectivity	Polarity_2	Polarity_3
		SVM	SVM	SVM
News Reviews	Baseline	**69%**	**58%**	**57%**
	With Cluster 1000	**69%**	**58%**	**57%**
Restaurant Reviews	Baseline	**71%**	**83%**	**73%**
	With Cluster 1000	70%	82%	72%
Market Reviews	Baseline	**89%**	88%	69%
	With Cluster 1000	**89%**	**91%**	**72%**
Movie Reviews	Baseline	**45%**	80%	52%
	With Cluster 1000	**45%**	**81%**	**52%**
News	Baseline	35%	**80%**	**71%**
	With Cluster 1000	**36%**	**80%**	**71%**

Table 4.24: Combine baseline model with cluster feature model at the document Level classification

		Subjectivity	Polarity_2	Polarity_3
		SVM	SVM	SVM
News Reviews	Baseline	**88%**	**56%**	**58%**
	With Cluster 1000	**88%**	**56%**	**58%**
Restaurant Reviews	Baseline	**96%**	**85%**	**67%**
	With Cluster 1000	**96%**	**84%**	66%
Market Reviews	Baseline	93%	90%	70%
	With Cluster 1000	**94%**	**92%**	**72%**
Movie Reviews	Baseline	NA	80%	44%
	With Cluster 1000	NA	**81%**	**45%**
News	Baseline	**63%**	76%	**65%**
	With Cluster 1000	**63%**	**77%**	**65%**

Table 4.25: Some of the words in same clusters of restaurant review domain

Custer ID	Word
1010101111011	(**هبطت** / *hbTt* / 'landed')
1010101111011	(**إنخفضت** / *InxfDt* / 'decreased')
1010101111011	(**سييء** /*syy′*/ 'bad')
1010101111011	(**مزحوم** /*mzHwm*/ 'crowded')
1010101111011	(**هادي** /*hAdy*/ 'quiet')
1010101111011	(**متواضع** / *mtwADς* / 'humble')

The only issue with this method was with the restaurant review domain. The result of polarity classification process was not improved by merging the cluster ID with the word. From this we infer that the domain of the restaurant review has more overlap between the words that are used in this domain. In addition, the cluster was not able to preserve the sentimental orientation of the words within the same cluster. Table 4.25 represents some of the words of the restaurant domain and shows that there are some different sentimental words within the same cluster. Most of the words in this cluster have a negative orientation such as (**سيئ** /*syŷ*/ bad). Some other words such as (**هادي** /*hAdy*/ quiet) carry positive meaning but are found in a same cluster that has mostly negative words. This behavior might affect the classifier on the opposed feature model for the restaurant domain.

4.9. Chapter Summary

This chapter gives a comprehensive investigation of document- and sentence-level Arabic sentiment classification. It also contributes to Arabic sentiment analysis by proposing new features that are added during the process of classification. The first section of this chapter shows the method of sentiment analysis that we follow depending on the machine learning based approach. The details of the general approach, primary feature models, and ML classifier are discussed in the beginning part of this chapter. With the primary and some of the advanced feature models, we use two different state-of-the-art ML classifiers, MNB and the SVM with linear kernel as well as nonlinear one. The second part of this chapter illustrates the advanced features that we used in the Arabic sentiment classification problem. It also focuses on explaining how the details of our new proposed features help during Arabic sentiment analysis. These models include polarity based and word clustering based features. The process of sentiment classification in the English language achieves a varied performance ranging from 58% to 97% accuracy (Pang and Lee, 2008; Turney, 2002). Our experiment achieves a comparable performance for Arabic sentiment analysis.

The subjectivity classification in Arabic sentiment analysis is a hard process than the polarity classification. This intuition comes from the results of the classification process that are performed in the experiments section. This also similar to what we have in another language such as English (Pang and Lee, 2008). In Subjectivity, the document's parts are included during the classification process. This would add extra information that may or may not useful to the classifier. In addition, some of subjective text (words of phrases) could be in the objective document. This also would make the distinguish process between the two classes (objective and subjective) is hard. Therefore, more work needs to be done here.

The baseline experiment shows that the primary BOW model plays the main role in Arabic sentiment analysis. Among all dataset domains and different classification level and types, this model achieves reasonable F1-scores. In addition, it illustrates that the SVM outperforms the MNB in most cases. Adding different n-gram models does tend to improve the SVM results, and it can add some increases in the MNB classifier at the sentence level classification. However, the performance of the SVM classifier increases using different n-gram models in the case of long text in a document level classification. This might indicate that the different n-grams can capture more relationships between words in the long text better than the small text.

Adding linguistic features to Arabic sentiment analysis helps to improve the performance of the classifiers. It increases the performance up to 9% in some cases where the typical variation in F1 over 5-fold is about 4%. "POS and ADV_ADJ" work better with the MNB than the SVM. However, they have some significant impact on the SVM in some cases. For example, the performance of the SVM improves by 5% in the event of polarity_3 sentence level classification. In addition, adding BPC feature to the model tends to increase the performance. The additional knowledge includes new relations between words that are not retained using the POS. The words have different sentiment orientation when they come in a different phrase. Therefore, the BPC was introduced to add this relationship between words to the classifier. It helps the classifier sometimes by increasing the accuracy up to 5% in the document level classification with long text domains.

The stylistic feature has a small impact on the process of classification especially in the subjectivity classification. However, it has some benefits to another classification type in the case of long text domains. We have noticed the performance was increased using this

feature in a document level classification with the newswire domain. This might indicate that the stylistic feature could improve Arabic sentiment analysis in the case of long formal text. Regarding the other stylistic feature that includes the position the sentence in the document, we find that approach might work in long text to eliminate unnecessary text that might not contribute to the overall sentiment of the text. The result of the classifier was improved in the case of two long domains that are movie reviews and newswire.

The last two proposed methods in Arabic sentiment analysis were using the polarity concept as well as the word cluster ID. We illustrate some of feasibility of using these approaches with Arabic sentiment analysis. There is not significant improvement in the results compared to the Baseline model, but we noticed that there are some slight improvements. These methods do not add more space to the feature model. The performance of the classifier has not been hurt significantly in most cases. Therefore, those features might have a significant impact on Arabic sentiment analysis. The drawback of the polarity method is that the Translation Machine we used might cause a slight improvement to the classifier using this method. However, it demonstrates that the translation from one language to another may preserve the sentiment orientation of the original language. In the case of the word clustering method, the word clustering technique need lots of data to capture the correct clustering between words. Therefore, using the same domain for performing the word clustering method may affect this approach. Finally, these two proposed approaches seem beneficial to be added during Arabic sentiment analysis, especially with the Dialect Arabic type that has an absence of basic NLP tools.

From all experiments that we did in this chapter, we may make general suggestions and thoughts. In the case of short type domains, such as news reviews and market reviews,

the best features which work well with them are morphological features, (POS or Adj_Adv, BPC) and the word cluster feature. The MNB classifier works better in this type of domain. The other features and classifier do not add much performance in this type of data because of shortness of its structure.

In the medium type domain, such as restaurant reviews, the BPC feature plays a major roles in the improvement of the sentiment classification process, especially in polarity_2 classification. In addition, the MNB tends to perform better than the SVM. The feature that counts the number of sentences in the document may be suggested to use in this domain during the polarity_2 classification. Lastly, using the Adj_Adv feature with this type of domain helps the process of sentiment classification in both classification levels and types.

Regarding long type domains, such as movie reviews and news text, using different n-gram models helps to capture the relation that might be found in the long text. As a result of that, the performance of sentiment analysis increases, especially in the document level classification. Because the short type domain benefited from the morphological features, the long domain gets the same benefits of using these features. In sentence level classification with these domains, the number of words feature increases the performance as well as the number of sentences. The sentence position would be the best feature and may play the best role in Arabic sentiment analysis of this type of domain. The polarity feature also works well in this type of domain, especially in the case of polarity_3 classification with movie reviews. The cluster feature tends to works with subjectivity classification better, especially in News texts.

Generally, the SVM works better with the following features: n-gram models in (subjectivity or polarity_2) document level classification, morphological features in polarity (sentence or document level) classification, position feature in subjectivity document level classification, and the cluster feature in polarity_2 (sentence of document level) classification. The MNB works better with BPC or Adj_Adv feature in polarity_2 document classification and the stylistic feature works best in (subjectivity and polarity) (sentence and document) level classification.

From all experiments described in this chapter, we can come up with some generalization about the feature models. In the sentence level classification, the best feature is "Adj_Adv". This feature used the adjective and adverb parts of the speech tag with the baseline model. The other possible features are BPC, word cluster, and polarity feature. In subjectivity sentence level, it seems the polarity feature works better. The BPC or the word cluster features also tend to perform well in the Polarity_2 classification. In Polarity_3, the Adj_Adv feature outperforms the others. In the document level classification, it seems that the n-gram features play the best role in the case of subjectivity classification. Using more n-gram, such as "bi- or tri-gram" may capture some relationship between the words in the document and help to discover some of subjective or objective aspect of the words or the phrases. The other two possible features that work better with the document level classification are the BPC and the word clustering features. The BPC helps to capture some sentiments orientation of the phrases that may be used by the classifier in the case of the polarity_2 and polarity_3 classification. In addition, the word clustering feature adds knowledge of groupings semantic of words. This knowledge also may infer some sentimental sharing between the words in the same cluster group.

In the end, we can make general thought about the different features sitting model's experiments. When the performance of the different classifier improves using a specific feature, this gives strong evidence that the particular features are robust and meaningful to Arabic sentiment analysis. Whenever the features do not improve the different classifier, this suggests that particular feature is not robust or relevant to Arabic sentiment analysis. Lastly, the features might be useful but less robust in cases of mixed performance results with different classifiers.

CHAPTER 5

Negation in Arabic Sentiment analysis

The previous chapter of this work focused on the different feature types that play roles in Arabic sentiment analysis in both levels (document and sentence) and types (subjectivity and polarity) of classification. In addition, they examine the effect of these different types of features on various ML classifiers, MNB and the SVM. Some of the main findings are figured out from previous works with features, such as the effect of using the morphological features with Arabic sentiment analysis. These features are not a completed list and there are different directions could be followed to improve Arabic sentiment analysis. One of these directions is the negation concept. Negation plays a central role in the sentiment of the text. Without caring about negation while sentiment is analyzed in the text may hurt the performance of the classifier. This chapter will show our proposed approach to deal with negation in sentiment analysis of the Arabic language.

The chapter starts by describing the problem of negation in sentiment analysis field. It is then explaining the negation concept in the Arabic language in Section 5.2. The chapter then depicts the importance of negation in opinioned Arabic texts in Section 5.3. In addition, the negation words list was devised to help in this task and other NLP application. The static method used is proposed in Section 5.4. The more sophisticated approaches to dealing with negation are explained in Section 5.4. The experiments and evaluations that were conducted in this work are shown in Section 5.5. The summary section ends this chapter by highlight the main points.

5.1. Introduction

Sentiment analysis of Arabic is still in its early stage as shown in Chapter 2. The most common linguistic aspect that affects sentiment analysis is negation. Negation often changes the sentiment orientation of a sentence. For example, the following two sentences, "this is a good movie" and "this is not a good movie", will have the same polarity when the negation item "not" is ignored in sentiment analysis. The positive sentiment associated with the word "good" is inverted into negative sentiment for the phrase "not good" and may not necessarily be as negative as the sentiment associated with the word "bad". Therefore, negation items and their scope in the sentence have to be taken into account during sentiment analysis (Wiegand et al., 2010).

Determining negation in a sentence is not an easy task due to the compound nature of negation. Negation words such as 'not' and 'no' do more than merely demonstrate negation in the sentence, but also possess further semantic meanings. The appearance of these words does not always indicate negation, particularly in the Arabic language. The negation words can in one instance be used to express negation and to express other meanings. In addition, the negation style can be expressed in sentence without using any of the negation words. In Arabic, negation may be expressed by using a wishing style such as (**ياليت هذا المطعم كان رخيصا** /*yAlythðAAlmTςmkAnrxySA*/ I wish if the price of this restaurant was cheap). In this sentence, the word 'cheap' can express positive polarity concerning the restaurant, due to the fact that it is cheap. However, the actual intention of the expression is the restaurant was not cheap. Hence this sentence conveys, in reality, a negative polarity.

Many other works study the effect of negation in detail in the English language while few Arabic studies touch this issue because this field is still at an early stage. Most of

the previous works also in Arabic sentiment analysis neither include the negation concept in sentiment analysis nor clarify the negation words list that they rely on. In addition, most of the works that include the negation theory use the semantic based approach to resolve the sentiment in Arabic text, not machine learning based approaches. The previous works that contain negation in Arabic sentiment analysis are illustrated in Chapter 2. The issues that relate to the previous works would be mentioned in this chapter. Firstly, the works did not mention the Arabic negation words used, stating only that they used around twenty words as negation words. Secondly, there is the issue of how they determined the negated words or phrase that come with the negation word in the sentence. This might affect the process of sentiment analysis since it has the possibility of changing the polarity (i.e. its polarity type and strength). In addition, relying on a simple representation (i.e. frequency counts of negation words or polarity words) would not capture all the semantics and syntax of the sentence in order to assist in sentiment classification. Some of the old methods (Hamouda and El-Taher, 2013) may work only for the domain chose, such as the posts and the comments in Arabic Facebook News Pages . This might, or might not, work with typical Arabic sentiment analysis.

This chapter focuses are explaining the details about using negation in the Arabic text sentiment classification. How does negation work in either Modern or Dialect Arabic and how does it cooperate with sentiment? Finally, what is the best method of dealing with negation in the case of Arabic sentiment analysis?

5.2. Negation in Arabic language

Negation in the Arabic language is used to negate the idea of the sentence. There are two styles of negation (Ryding, 2005; Wright and Caspari, 1898). The first style uses negation

terms, called explicit negation. The second style is implicit negation that does not use negation terms or words. Instead, some of the words or forms in a sentence carry a negation meaning. The scope of this work will be focused solely on one type of negation, explicit negation.

5.2.1. Explicit Negation. Explicit negation is a negation style that is used to negate the sentence using one of the negation words. The negation terms, tools, items, or words in the Modern Stranded Arabic are "لا، لم، لما، لن، ما، ليس، إن، لات، غير" (Wright and Caspari, 1898). Table 5.1 shows transliteration and the English meaning of these words and their types. The negation item, word, or terms will be used to express the negation words in this chapter interchangeably. The majority of these negation terms, apart from two, are considered to be prepositions. (ليس / *lysa* / Not) is deemed to be a verb and (غيرَ/ *γyra* / But) is regarded as a noun. Some of them also could be used with a nominal sentence, or with a verbal sentence, in order to negate the sentence. In addition, these negation words could appear first in the Arabic sentence, or in front of the verb or the adjective that is to be negated. The majority of these negation words are used mainly in Modern Standard Arabic (MSA), as well as in Dialect Arabic (DA) (Farghaly and Shaalan, 2009). DA has particular negation items that are used for a specific dialect. Negation words (مو / *mw* / No or Not) and (مِش / *mish*/ No or Not) are used in a number of particular dialects in order to express the same meaning as using 'لا/*lA* ' meaning 'not' or 'no'). These negation items are considered with the negation words list because it is used widely in DA that is in our corpus.

Since one of the negation words (ليس / *lysa* / Not), is a verb, it must be conjugated in order to suit different subjects (Ryding, 2005; Wright and Caspari, 1898). There are different

Table 5.1: Modern Stranded Arabic negation words

Arabic Negation word	Transliteration - English meaning	Its type
لا	*lA*- Not or No	
لم	*lm* - Not	
لما	*lmA*- Not	
ما	*mA* - Not	preposition
لن	*ln* - Not	
إن	*In* - Not	
لات	*lAt* - Not	
غير	*γyra* - But	noun
ليس	*lysa* - Not	verb

Table 5.2: Different forms of the negation term *lysa*

Singular	(ليس /*lysa*)	Singular
	(لست / *lsta*)	Singular male
	(ليست / *lyst*)	Singular female
Dual	(ليسا / *lysA*)	Dual
	(لستما / *lstmA*)	Dual male
	(ليستا / *lystA*)	Dual female
Plural	(لسنا / *lsnA*)	Plural
	(لستم / *lstm*)	Plural male
	(لستن / *lstn*)	Plural female
	(ليسو / *lysw*)	Plural
	(لسن / *lsn*)	Plural female

forms of this word that must agree with the subject in terms of both gender and number. In Arabic, there are three types of quantity names: singular, dual, and plural. Table 5.2 demonstrates these different forms of the word (ليس / *lysa* / Not).

A number of negation terms are used not only for negation: they may also be used to change the style and semantic meaning in Arabic. The negation item (ما /*mA* / What) may be used in various ways, such as in condition, interrogative, and wondering. For example, (ماهذا الذي تقوله ؟ / *mA hðA Alðy tqwlh*? / what are you saying?). In this sentence, the word (ما /*mA* / What) is used to express the question, rather than to negate the sentence.

5.2.2. Implicit Negation. Implicit negation is a style of negation that does not use negation terms. The negation can be achieved using interrogation, condition, and wishing styles. These styles are used in metaphorical ways in order to reflect the meaning of negation instead of the actual meaning of the style.

In the interrogation style, the primary aim is to express negation and not to put forward a query. For example, (من يشترى هذه الكاميرا الا المضطر / *mnyštrýhðhAlkAmyrAAlAAlmDTr* / Who want to buy this camera except the one who is in need?) the actual intended meaning in Arabic is "no one buys this camera except the one who is in need". The last implicit negation style is the one using the wishing mean. In this style, the negation is expressed by the style itself, as the wishing concerns asking something that would not happen. For example, (اتمنى لو كان هذا المطعم رخيصا / *Atmný lw kAn hðA AlmTςm rxySA* / I wish if this restaurant was cheap) this carries the negation of the main adjective in the sentence, which is cheap. Therefore, the actual meaning of this sentence is that the restaurant was not cheap.

5.3. Importance of Negation in Arabic Sentiment

This section illustrates the importance of using negation to express sentiment in Arabic text. This importance may be expressed by showing the percentage of the opinioned sentences that have negation words. The datasets that are used in this analysis are discussed early in detail in Chapter 3.

In order to compute the percentage of the negation in each dataset of the corpus, it is necessary to specify the negation words listed first, and then the method in which the negated sentence should be determined. Depending on the investigation into the Arabic grammar rule concerning negation words, there are around twenty negation words, including

Table 5.3: Percentage of the negation at the document and sentence level for each data domain

	Percentage of the negation	
Data Type	Document	Sentence
News reviews	55%	13%
Market reviews	99%	19%
Restaurant reviews	39%	11%
Movie review	14%	11%
Newswire	66%	8%

all morphological forms of '**ليس**/*lysa*'. Tables 5.1 and 5.2 show these words and their types. The majority of them are used both in MSA and DA. We also add two more DA negation words as explained early. These items are (**مو** / *mw* / No or Not) and (**مِش** / *mish*/ No or Not).

The typical method of writing these negation words in Arabic is by adding a space before and after. This method will be considered while the negation is counted in the sentence. Any sentence that has a negation words is counted as a negated sentence. This counting is achieved on two levels: document and sentence. When it comes to the document level, any document is counted if it contains negation words. This is also applied on the sentence level. Table 5.3 shows the percentage of the negation in our dataset. The second column of Table 5.3 displays the percentage of documents containing negation words for each dataset. For example, there are around 759 documents in the restaurant dataset that contain negation words, this being approximately 39%. It is also noticeable that almost all movie reviews include negation terms. The third column of Table 5.3 demonstrates the percentage of the negation at the sentence level. Between the tenth and the twentieth sentence of the corpus there are negation words which may play the key role of flipping the sentiment orientation of the sentence. This might effect the process of the classification if negation is ignored.

Table 5.4: Negation items groups depending on percentage of its appearance in the corpus

First group	Second group	Third group
(lA / لا)	($lysa$ / ليس)	(In / إن)
(lm / لم)	(γyra / غير)	(lmA / لما)
(mA / ما)	(ln / لن)	(lAt / لات)
	(mw / مو)	
	($mish$ / مِش)	

Table 5.4 arranges the negation words from the most frequent to the least. Depending on that, these words may be classified into three categories. The first category is the one used most frequently. The second is the one often used . The last is used either rarely or not at all, i.e., '**لات**/lAt ', which is not used in a sentence in our dataset, having been used in Classical Arabic (CA) but neither in MSA nor in DA.

Table 5.5 depicts the percentage of sentences that contain any of the negation words for each dataset in each class, either objective or subjective. For all objective or subjective sentences, the sentence in this class is counted if it has a negation word. It is clear that negation words tend to be used more in a subjective class sentence in order to negate the sentiment orientation of the sentence in the three datasets. In the news dataset, it appears that the negation is frequently used in the case of objective sentences, due to the fact that the majority of sentences in this domain may carry factual information. Therefore, the negation words here are used mostly to negate factual information rather than reverse the sentiment orientation of the sentence. In addition, the greater usage of these words in both movie reviews and news domains may arise from their different usage. As previously explained, these negation words could be used in other styles rather than to negate the sentence.

The same study is undertaken for the polarity classes. Table 5.6 shows the percentage of the sentences that contain negation words in each class for each dataset. The bold text in

Table 5.5: Percentage of negated sentences for each dataset in each class that is either objective or subjective

Dataset	Objective	Subjective
News reviews	25.1%	**64.9%**
Market reviews	06.7%	**83.3%**
Restaurant reviews	32.8%	**57.2%**
Movie review	**48.7%**	31.3%
Newswire	**56.1%**	23.9%

Table 5.6: Percentage of negated sentence in each class of polarity for each dataset

Dataset	Positive	Negative	Neutral
News reviews	16.9%	**34.6%**	13.4%
Market reviews	26.2%	**39.9%**	17.3%
Restaurant reviews	19.3%	**30.8%**	07.1%
Movie review	10.3%	**11.2%**	09.8%
Newswire	08.7%	**12.9%**	02.2%

this table highlights the highest percentage among the three classes. It appears that negation is frequently used in the case of the negative class. In addition, the negation words tend to be used more in the case of the positive or negative polarity classes, when compared to the neutral one.

The final investigation that has been undertaken with the negation is displayed in Table 5.7. This attempts to establish the percentage of the negation that has occurred after the classification process at the sentence level. In this investigation, the classification processed for each categories, subjectivity and polarity, is based on the SVM classifier with the uni-gram feature model. The numbers of negated sentences that are incorrectly or correctly classified are then counted. For example, there are around 108 negated sentences with 32.34% in the error classification, whereas, there are 162 negated sentences with 22.34% in the correct part of subjectivity classification for movie reviews dataset. This implies that there are more negated sentences that are incorrectly classified then correctly classified, and the classifier

Table 5.7: Percentage of the negation that has occurred After the sentiment analysis process

	Subjectivity		Polarity 1		Polarity 2	
	Percentage of negated sentences are classified					
Dataset Domain	correct	incorrect	correct	incorrect	correct	incorrect
News reviews	**25.07**	20.94	29.13	**30.99**	29.51	**29.57**
Market reviews	**30.55**	24.10	30.00	**34.48**	28.78	**43.54**
Restaurant reviews	19.50	**30.02**	18.40	**24.89**	19.10	**26.60**
Movie review	22.34	**32.34**	22.75	**48.78**	27.88	**45.89**
Newswire	09.41	**20.30**	17.97	**18.92**	**17.60**	16.00

needs to be aware of negated sentences during the classification process in order to avoid this issue. By looking at Table 5.7, we found that there is a higher percentage of negated sentences that are incorrectly classified than correctly classified in different classification categories among all datasets. This suggests the importance of the negation words and their effect on other words or phrases. Therefore, Arabic sentiment analysis should address this issue during the classification process.

5.4. Proposed Method to Handle Negation

In this work, it is suggested that the work should be carried out in two areas in order to employ negation in Arabic sentiment analysis. The first area depends on a simple method of discovering negation. For the second area, more complex models will be relied upon in order to establish how to determine negation in Arabic sentence before the sentence is processed for sentiment analysis.

5.4.1. Primary or Static Approach. In this type of approach, we rely on primary methods for defining and injecting negation while processing sentiment in Arabic text. In order to deal with negation in sentiment analysis, two aspects should be determined. These aspects discover the negated sentence and capture the actual scope of the negation in the

sentence. To determine the negation in the text, we assume that whenever the negation items are found in a sentence we consider this sentence as a negated one. This assumption is not true at all, but it gives us the first step toward adding negation in Arabic sentiment analysis and helps us to figure out if the negation plays a major role in the sentiment classification process. We call this a static approach because of the primary method of negation detection in the text. Because some of the negation words in Arabic are used to express other writing styles rather than negation, it is proposed not to use all negation words on the list to determine whether or not a sentence is negated. Only the most common negated words will be relied upon; i.e. those that are most often used for negation. We rely on the first and second groups that are shown in Table 5.4.

After capturing the negated sentence, the scope of the negation should be specified. Different methods have been proposed. The first method assumes that the negation item only affects the word after it. The second one applied different windows sizes to capture the scope of the negation such as 2 or 3 words after the negation item. The third approach assumes that the whole sentence is negated that means all words after the negation word in the negated sentence.

The other feature to be added in this area will be the stylistic feature of negation. It is proposed to count the number of negations found in the text. This will add some useful information to the classifier in order to inform it about a negation that takes place in the sentence or document.

The last method depends on the Base Phrase Chunk (BPC). In a natural language processing, BPC is a process that separates and segments a sentence into its phrases such as noun, verb, or prepositional phrase. BPC represents a shallow parser tree of the sentence.

This method will be depended upon BPC in order to determine the scope of the negation by assuming all words in the same phrase with the negation word are negated. In this method, all words either in the phrase that has negation or the next phrase after the negation phrase should be assumed negated.

The next step is how the negation concept would be injected with sentiment analysis in Arabic text. For each previous method, we assume the negation could be injected while building the feature model. For each negated word, the artifact tag "NOT" should be attached to that word during building the feature model. If the word x is in the scope of the negation item, then the new feature word will be added to the vector space model, which is "x _NOT". This method follows the one undertaken in the English language by Pang, et al., (2002). After that, the process of the classification will be performing as explained in Chapter 4. The different mechanisms of adding this negation static awareness will be evaluated in the experiment section.

5.4.2. Dynamic or complex proposed approach. The previously proposed model provided details of how negation can contribute to Arabic sentiment analysis process by proposing features that distinguish the negated from non-negated words. It relies on a simple, static method of determining negated sentences, or even the scope of the negation in a sentence. There are two important factors in the negation concept. First, how can it be determined whether a sentence is negated or not? Second, a method is required to provide information about the scope of the negation in a sentence. Much research relating to discovering negation and its scope has been done on other languages (Wiegand et al., 2010), but no study has touched this concept in the Arabic language. Therefore, a method

is proposed that first deals with these issues of negation before adding its effect to sentiment analysis in the Arabic language.

In order to work in this area, annotated corpora for negation in Arabic text are required. To the best of the authors' knowledge, there is no specialized corpus for Arabic in this domain because the field of sentiment analysis is relatively new in Arabic compared to other languages. Therefore, work needs to be carried out in order to enrich the research community with a negation corpus. The same procedure should be followed as that followed while building the sentiment corpus. All negated sentences will be annotated, and the negation words that are used will be also annotated along with the scope of this negator. The sentence: "I do {not} [like playing football] but walking through the street" is an example of a negated sentence with annotation scheme. The negator is the "not" word surrounded by "{}", and the brackets determine the scope of the negation, "[]".

In other languages, the negation tasks are considered to be the same as any other labeling sequence tagging problem, such as the POS or the name entity recognition (Councill et al., 2010; Wiegand et al., 2010). After the process of annotation is complete, the labeling algorithm "Conditional Random Field" (CRF), which is used in different tasks, will be used. This has had broad applications in natural language processing, computer vision and Bioinformatics (Sutton and McCallum, 2011). It is also used in the English language to determine the scope of negation, using annotated negation corpora by Councill, et al.,(2010). After training CRF on the annotated negation data, the output of the CRF is used to guide the sentiment analysis in order discover a negation and its scope. Figure 5.1 shows how this proposed method cooperates with sentiment analysis. The detail of the performance and evaluation of this method will be explained in experiment section.

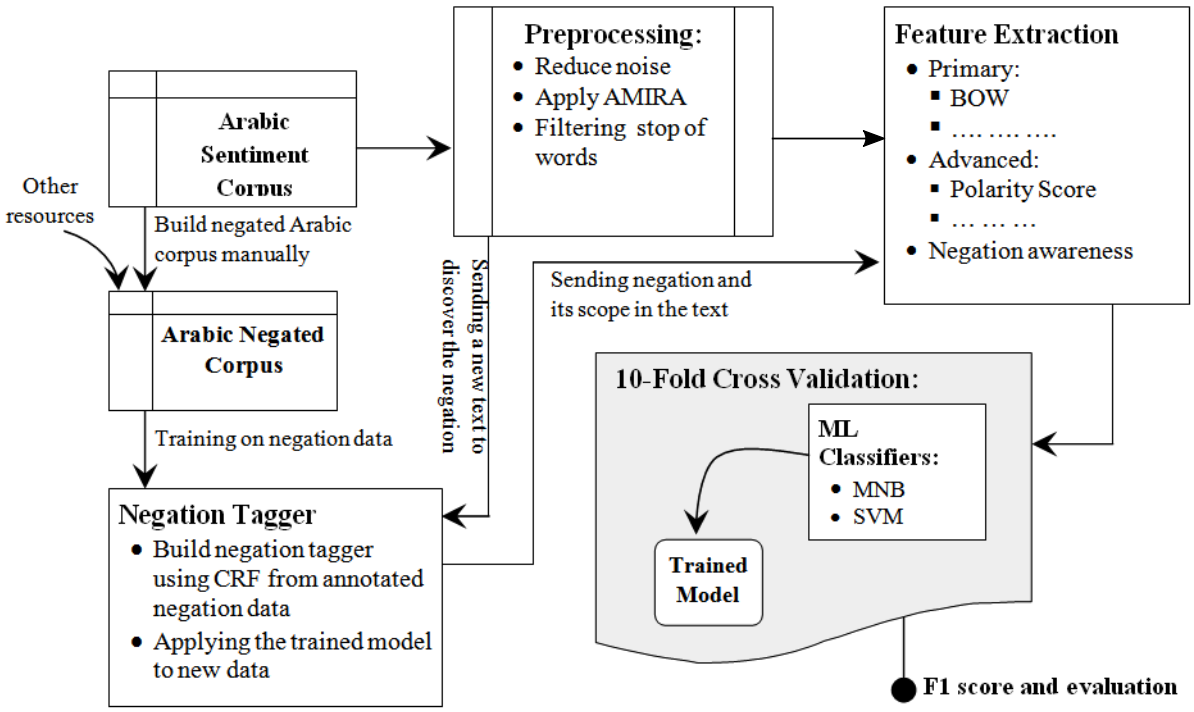


Figure 5.1: Dynamic approach to capture negation and its scope in Arabic sentiment analysis

5.5. Experiments

This section explains the experiment setup that we following during the evaluation process of adding the negation concept to Arabic sentiment analysis. The first part explains the experiment setup that we follow. The details of the experiment of each approach and its results are also discussed in the following sections.

5.5.1. Experiment setup. The process of adding negation is performed on different domains of sentiment data in Arabic text. It also includes different types of Arabic language, MSA and DA. We use the same sentiment corpus that we built in order to apply the negation in the problem of sentiment analysis in Arabic. The details of this corpus are explained in Chapter 3.

To evaluate the negation awareness in sentiment analysis for Arabic text, many experiments were undertaken using the SVM with linear kernel. As a basic step, the uni-gram model is applied for the learning and testing process. We relied on the scikit-learn library (Pedregosa et al., 2011) for using machine-learning classifiers. ARMIRA (Diab, 2009) is used to get the BPC of the Arabic text. The classification process is achieved on the sentence level only since the negation is more related to the sentence than the document. We suppose that whenever the sentence level classification process is improved then the document level should be improved too. In the experiments, that test proposed techniques in this chapter, we follow these steps:

- Whenever a negation item is found in a sentence, the sentence is considered negated with that item. This is not applicable with regard to the dynamic approach.
- To build feature model used by classifiers, a uni-gram model is used that takes the distinct words within the dataset.
- The artifact "NOT" tag was added to all negated words after the negation item in negated sentences. If the word x is preceded by the negation words, then the new feature word will be added to the model, which is x _NOT. This method follows the one undertaken in the English language (Pang et al., 2002).
- The proposed approaches differ based on which word should have the "NOT" tag in the feature vector.
- 5-fold cross validation is performed to test proposed methods.

To evaluate these classifiers, we rely on calculating F1-score. This evaluation metric depends on the two other metrics that are precision and recall. In order to compute these metrics,

the confusion matrix should be generated after the classification process. We follow the same method of evaluation metric that is illustrated in Chapter 4.

5.5.2. Experiments with Static approach. The first experiment investigates the effect of adding negation words to the feature model. Most of the research in regard to Arabic natural language processing considers negation words as a stop word that should be removed before building the feature model. This may work well with some natural language processing problems, but it does not work for sentiment analysis. Therefore, we compare two feature models: one that considers negation words as stop words and eliminates them from the feature model baseline "*NoN: No Negation*". The other feature that includes negation words within the model, called "*WtN: With Negation*". The uni-gram model is used to build these feature models. Table 5.8 illustrates the results of classification using the SVM with these models. The first row in each dataset domain represents the first feature model, which is "*NoN: No Negation*". The second one shows the second feature that includes negation words with the feature model and removes stop words "*WtN: With Negation*". The values display the average F1-score of performing the SVM within the particular feature type. The bold numbers display the best value achieves with a specific approach.

In Table 5.8, we notice that the results are better when the second model "*WtN: With Negation*" is used, which includes negation words with the model. This is correct for a different types of classification, subjectivity, polarity_2, or polarity_3. It is clear that negation words play a role in sentiment analysis because they flip the sentiment orientation of the sentence. Additionally, the performance of the classifier increases between 1 and 2 percent in most cases. In some cases, the difference between these models is as large as 8 percent, such as in the case of polarity_2, the second column in Table 5.8 with the movie review dataset.

Table 5.8: Performance of the primary negation method on sentence Arabic sentiment classification

	Feature	Subjectivity	Polarity_2	Polarity_3
News Reviews	NoN	69%	58%	57%
	WtN	71%	58%	57%
	WW1	72%	58%	**58%**
	WW2	74%	59%	**58%**
	WW3	72%	59%	57%
	WS	74%	58%	57%
	NC	74%	**60%**	**58%**
	BPC	**75%**	57%	56%
Restaurant Reviews	NoN	71%	83%	73%
	WtN	73%	84%	74%
	WW1	74%	85%	74%
	WW2	74%	86%	75%
	WW3	74%	86%	75%
	WS	74%	86%	74%
	NC	74%	86%	75%
	BPC	**75%**	**87%**	**76%**
Market Reviews	NoN	89%	88%	69%
	WtN	**90%**	**90%**	70%
	WW1	89%	**90%**	70%
	WW2	89%	**90%**	**71%**
	WW3	89%	**90%**	**71%**
	WS	89%	**90%**	**71%**
	NC	89%	**90%**	70%
	BPC	89%	84%	**71%**
Movie Reviews	NoN	45%	80%	52%
	WtN	46%	89%	56%
	WW1	47%	89%	56%
	WW2	**48%**	90%	57%
	WW3	**48%**	90%	57%
	WS	**48%**	91%	**58%**
	NC	**48%**	90%	57%
	BPC	**48%**	**92%**	**58%**
Newswire	NoN	35%	80%	71%
	WtN	36%	80%	**72%**
	WW1	37%	80%	**72%**
	WW2	39%	**81%**	71%
	WW3	37%	**81%**	71%
	WS	**40%**	79%	70%
	NC	**40%**	**81%**	71%
	BPC	39%	80%	**72%**

Key: "NoN" No Negation, "WtN" With Negation Words, "WW1" Window of one negated Word, "WW2" Window of two negated Words, "WW3" Window of three negated Words, "WS" , "NC" Negation Counting, "BPC" Base Phrase Chunk, all words in the same phrase where the negation word is located

Table 5.8 illustrates the different window sizes in order to capture the effect of negation and compares their results. As explained earlier in section 5.4.1, the artifact "NOT" tag will be added to the negated word while the feature vector is built. The one word window will include one word after the negation word in order to add "NOT" to the word. This situation refers to "*WW1: Word Window 1*", which is the third row in Table 5.8. The row "*WW2: Word Window 2*" uses two words after the negation item, which is shown as fourth row. The fifth row displays "*WW3: Word Window* 3", which includes three words after the negation item. The "*WS: Window of Sentence*" row includes all of the words after the negation item until the end of the sentence.

Within the subjectivity classification, using these techniques increases the results by at least one percent in most cases, except in the case of the market review. The length of the sentence may affect these techniques, as shown by the market review having the shortest sentence length compared to other datasets. Another explanation may be in regards to the assumption that any sentence having negation words would be negated. This situation may not work well in this domain because some of the negation items may be used in a different style than to negate the sentence. In the case of the polarity classification, the results are similar to the subjectivity classification. The best window size is when two words are used after the negation item or using all words after the negation. The performance increases by one to two percent. This different window size helps also in the case of polarity_3 classification in regard to the movie reviews. The result increases to 58% from 52%.

Table 5.8 also displays the result of the experiment examining the effect of using the stylistic negation feature in regard to the classifier. This method is "*NC: Negation Counting*" that adds the number of negation words in the feature model. This feature helps the classifier

by increasing its performance by two percent, especially in a medium or long sentences. In the case of short sentences, this technique does not help and it decreases the result in the case of the subjectivity classification. This might not contribute to capturing the real effect of the negation that flips the sentiment orientation of the words. However, it might work in the case of long sentences when there is more than one negation in the same sentence. Additionally, this scenario may work better in the case of document level classification because it would capture the number of negations in the whole review.

It seems that the *BPC* technique achieves good results in most instances, last row within each dataset Table 5.8. In the case of subjectivity, the performance of the classifier with the "*BPC: Base Phrase Chunk*" feature model method is improved. The F1-score increases by 2% to 4% percent across some of the datasets. The most interesting results are those of the polarity classification in Table 5.8. In the case of the long sentence in movie reviews and medium sentences in restaurant reviews , the result increases dramatically. For example, the results goes from 80% to 92% in the case of movie reviews. Additionally, it increases from 83% to 86% in restaurant reviews. This may be due to using Base Phrase Chunk to capture the scope of the negation in the sentences. This technique may work better in capturing the polarity than in the subjectivity classification.

On the other hand, the results decrease by 4% in the case of market reviews. This may be due to the nature of the dataset. The market reviews have the shortest sentence length compared to the other datasets. Additionally, more investigation should be done in order to find out other reasons for why this technique does not work well in regard to market reviews. It also suggests the the problem may come from using the AMIRA tool that only works well with the MSA, whereas the nature of the market reviews is Dialect Arabic.

Table 5.9: Average F1-score across all data domains for different static negation approaches

Feature	Subjectivity	Polarity_2	Polarity_3
NoN	62%	78%	64%
WtN	63%	80%	66%
WW1	64%	80%	66%
WW2	**65%**	81%	66%
WW3	64%	81%	66%
WS	**65%**	81%	66%
NC	**65%**	**82%**	66%
BPC	**65%**	79%	**67%**

Key: "NoN" No Negation, "WtN" With Negation Words, "WW1" Window of one negated Word, "WW2" Window of two negated Words, "WW3" Window of three negated Words, "WS" , "NC" Negation Counting, "BPC" Base Phrase Chunk, all words in the same phrase where the negation word is located

5.5.3. All static approach together. In static methods, the last experiment compares all of the techniques to find which method worked better in regard to the datasets. In order to compare these methods, we calculated the average of each feature results for all of the datasets. This is done for each classification type, subjectivity, polarity_2, and polarity_3.

Table 5.9 describes the result of comparing all of the static techniques that have been used in regard to negation within Arabic sentences while building the feature vector. The values of this table show the results for each type of classification over all dataset domains. Each row represents the result of using the particular technique. For example, the second row indicates the results of the classifier for each type of classification using the "*WtN*" model that includes negation words, and removes stop words. This feature model achieves 63%, 80%, and 65% average F1-scores in regard to subjectivity, polarity_2, and polarity_3 classification, respectively.

Table 5.9 tends to confirm the fact that negation plays a large role in Arabic sentiment analysis. The worst results are recorded by using the baseline model "*NoN*" that does not include negation items or its effect in the feature model. On the other hand, other methods

that use negation help the classifier and lead to better classification. Additionally, it is clear from Table 5.9 that the best feature model that works well to capture the negation orientation in regard to Arabic sentiment analysis is *BPC* (Based Phrase Chunk). For example, it achieves around 65% and 67% F1-score in the case of subjectivity and polarity_3. It makes sense that the negation would affect all of the words in the same phrase chunk or the following phrase. The negation counting method achieves best result in the case of polarity_2 classification with more than 82%.

5.5.4. Experiments with Dynamic Approach. This method needs the annotation process to annotate data in regards to the following two concepts: negation item and negation scope, as explained earlier, Figure 5.1. Due to this process being time consuming and taking a lot of effort to build the annotation corpus for negation in Arabic, as well as we do not have any previous work on this field, so we had to start with the initial steps to prove the concept of this method. Therefore, a small number of sentences are annotated with the negation concepts and scheme, as explained in Section 5.4.2. Around 50 sentences in each class were chosen.

The next step of proposed approach after the annotation process should be using the trained sequence-labeling algorithm on these datasets to examine its ability to capture negation in Arabic sentences. The output of this step will be used during building feature model that are used to train the SVM to analyze sentiment within Arabic sentences. In order to prove the concept of this method in regard to SA in Arabic, we assumed that this step had already been done and that the output would be the same as the annotation part. Therefore, the output will be used to judge if the sentence is negated and to capture the actual scope of negation by adding the "NOT" tag to each word in the scope of negation while building

Table 5.10: Results of using dynamic approach with static method

	Feature	Subjectivity	Polarity_2	Polarity_3
Restaurant Reviews	WtN	**85%**	59%	38%
	WW2	**85%**	59%	38%
	DyN	82%	**60%**	**54%**
Movie Reviews	WtN	**85%**	51%	37%
	WW2	**85%**	50%	38%
	DyN	83%	**61%**	**40%**
Newswire	WtN	**81%**	76%	65%
	WW2	**81%**	76%	**66%**
	DyN	80%	**78%**	59%

Key: "*WtN*" Include negation words with Uni-gram feature model, "*WW2*" Two Word Window, assuming two words are negated after the negation word, and "*DyN*" Dynamic Method depending on manual negation annotation and negation tagger.

the feature vector within each sentence. This method is compared with two of the static methods, which are the baseline that include negation items "*WtN*" and the window size of two words "*WW2*". The results would be different in the case of comparing this experiment with static methods because this test was performed on only a small number of sentences.

Table 5.10 illustrates the result of using the dynamic method with a small part of three different datasets (movie review, restaurant review, and news text). The classification process is performed at the sentence level. The bold numbers show the best result that is achieved by using a particular method in each classification type. The first row displays the baseline model of this experiment that includes negation words with the feature model. The second row illustrates the method of adding artifact "NOT" tag to two words after the negation items. The last row is the result of using dynamic approach "*DyN*".

Unfortunately, the result gets worse in the case of the subjectivity classification, but the result gets better in the case of the polarity classification. In subjectivity, the result decreases by 2% to 3% percent using a dynamic approach in comparison with the other methods. This could be due to the amount of data being used to test this approach.

In the case of the polarity classification, the results are much better for this approach in comparison to the other methods. In the movie reviews, the classification result increases from 51% to 61%. This might reflect the effect of actual annotation of the sentence with the negation concept. In the other two datasets, the result increases by 1% or 2% percent. The difference may come from the nature of each dataset. For example, negation might be used more in the movie review, so the annotation process may capture the negated sentence that actually plays the main role in sentiment analysis. On the other hand, other datasets, such as news texts and restaurant reviews, may use negation at a moderate level or use shorter sentences in comparison to the move review. Therefore, the fixed approach could be enough to capture the negation effect, but the manual annotation may add some improvement to that section. In the polarity_3 classification, the results are different. The dynamic method works better in medium and long sentences in the restaurant review dataset, but it does not improve the performance of the classifier in the newswire datasets. This may be due to the nature of the dataset. Neutral sentences may add some difficulties and ambiguities to the classifier. Additionally, negation may appear less in neutral sentences than in positive or negative sentences. Therefore, capturing the actual negation of a sentence does not help when adding the neutral class to the other polarity classes in Arabic sentiment analysis.

5.6. Chapter Summary

This chapter provides the first step toward handling negation while the sentiment is analyzed in the Arabic language. Most of the previous research does not include the concept of negation in regard to Arabic sentiment classification that uses the machine learning method. Even though some previous works handle negation in Arabic sentiment, they do not rely on ML based methods in the sentiment classification. The previous works only consider

the basic negation concept that might not capture the actual effect of the negation in the sentence. Therefore, the work in this chapter tries to leverage the performance of Arabic sentiment classification by injecting the negation effect into the ML base method.

Most research of the Arabic in natural language processing assume that the negation items are considered as the stop words that have not representation effect on the text. Therefore, these types of words are removed before the analyzing process. In the case of sentiment analysis, the negation words must be included in the analyzing process. Therefore, the model that includes these words achieves the best result compared to the baseline model that does include them. The first idea of our method is the assumption of a negated sentence is a sentence that has negation item. The second idea in this method is to determine the actual negation scope in the sentence. Different mechanisms are proposed starting with one word after the negation item until the ending of the sentence. The BPC provides a more intelligent method to specify the scope of the negation. We apply the negation tag to the words in the same or next phrase where the negation item is captured. Lastly, we proposed a method that relies on the stylistic feature approach by counting the number of negation found in the sentence.

The static methods show improvement in the performance of the classifier compared with the baseline model that does not have any awareness of the negation. There is no silver bullet that can solve every situation in every scenario. In some cases, the feature model that includes negation items achieves the best result. In other cases, we found adding the "NOT" tag to the negated words in the feature model helps more than other approaches. These variations depend on the nature of the text and the sentence structure. However, the *BPC* seems to be the best approach that captures the scope of negation because the *BPC* capture

shallow syntax of the sentence that might preserve the relation between words. Another good method to use is Negation Counting "*NC*" or window size with two words "*WW2*" after the negation item in the case of an absence of the BPC tagger, especially with the Dialect Arabic.

We have also noticed that the percentage of the negation is high in the three domains, which are news review, restaurant, and movie reviews. These domains gain better performances in the sentiment analysis when the negation is added, especially in the subjectivity classification. In addition, the negation was found more in the positive and the negative classes than the neutral. Therefore, adding the negation feature helps more in the case of polarity_2 more than polarity_3 classification.In subjectivity classification, it seems that using BPC to capture the negation works better than the other features. For the polarity classification, using any negation method has the equivalent increasing in the performance for Arabic sentiment analysis. However, using BPC in this case seems the reasonable method.

The dynamic method aims to find out the model that helps to capture the actual negation sentence in the Arabic text, as well as its scope. This method should work depending on building a negation tagger. After the negation tagger model is built, it can capture the actual negated sentence and its scope in a new text and then create feature model depended on that. This approach needs manual annotation process to annotate data with the negation concept. This process takes much time and effort. Therefore, this work tries to prove this idea by annotating a small number of data and supposing this data is generated by the negation tagging process. The negation scheme tags are then used to add the negation artifact tag "NOT" to the real words in the negation scope. This method seems promising and outperforms two static approaches, "*WW2*" and "*WtN*". However, it behaves abnormally

in the case of subjectivity classification that may due to the small amount of data the used in these experiments.

These are not comprehensive or optimal approaches to resolving the negation problem in Arabic sentiment analysis. It could be considered as start point that help to improve the result classification of Arabic sentiment analysis. In addition, it may help to establish an Arabic negation tagger that may be used with other Arabic natural language processing applications. This is would be help to discover the other type of negation, implicit negation, in Arabic sentence. The advice may be generated here for the new researcher in this field to concentrate their work with building the negation tagger for the different types of Arabic language, MSA, and DA.

The improvements in this field are endless and could be from different directions. The next chapters will give two different directions to improve the Arabic sentiment analysis area. The first one is by applying nonlinear ML classifier. The other route will include learning sentimental knowledge from a different domain.

CHAPTER 6

Using Neural Networks in Arabic Sentiment Analysis

Chapter 4 discussed different feature configurations that might play essential roles in Arabic sentiment analysis. It also used state-of-the-art ML classifiers to perform the sentiment analysis in the Arabic language. These classifiers were MNB and the SVM (Pang and Lee, 2008). Most sentiment analysis research used the SVM because of the robust of its performance was across different situations. This trend also duplicates in the case of Arabic sentiment analysis. However, the other ML classifiers such as Neural Networks "NNs" are used and investigated in other languages such as English (Sharma and Dey, 2012). In the case of the Arabic language, most of the work has used the SVM (Abbasi et al., 2008; Abdul-Mageed et al., 2011; Alhazmi et al., 2013). The reason why most of the researchers do not use the NNs comes from the amount of time that the NNs takes to train. This time consumption might come from the performance of the old machine. Nowadays, the performance of the computer has improved significantly from the past. The NNs may come into account as a choice to analyze sentiment in the text. In addition, the performance of the non-linear kernel of the SVM may show that the Arabic Sentiment has some non-linearity aspects that may need to be captured. Therefore, we will use the NNs in Arabic sentiment analysis and compare its performance with the state-of-the-art ML classifier that has been used in the Sentiment analysis problem.

This chapter is organized as follows. The first section gives some of background information of using the NNs in the field of sentiment analysis. The Second section explains the

methodology of how the NNs classifier is applied with Arabic sentiment analysis. It also describes the structure and configuration of the NNs is used. Two different evaluation types are illustrated in the experiment section. The discussion of evaluation and comparison are depicted in the results and discussion section. This chapter concludes with the summary section.

6.1. Introduction

Language with rich morphology and high inflection such as the Arabic language may need a different ML classifier that deals with non-linearity problem. The SVM has a stable performance across different configurations of the feature model. This stability inspires us to try other variations of ML classifier.

Neural Networks (NNs) achieve good performances in other natural language processing tasks, such as text classification (Chen and Chiu, 2009; Dhande and Patnaik, 2014; Harrag and El-Qawasmah, 2009). Literature survey shows that there is not enough work done in sentiment analysis of the Arabic language using the NNs. It is clear that the SVM achieves good results compared to the MNB. This also has motivated the investigation and use of other ML classifiers, such as the NNs, in the task of Arabic sentiment analysis. Therefore, the NNs will be used in Arabic sentiment analysis and its performance compare with the SVM.

The Neural Networks concept has been used a lot in many research projects in other languages such as English. Sharma and Dey (2012) claimed that using the NNs achieves around 95% F1 score on a movie review domain. The classification process of that work was on polarity document level classification that included only positive and negative categories. Four different approaches, Information Gain (IG) and three sentiment lexicons, were used to

build the features. The traditional bag-of-word model was built, and the top n-ranked words the were used as a feature. They figured out that the IG mechanism achieved the best result with the NNs classifier. In addition, this method could be used to reduce the dimensionality of the space vector model and does not hurt the classification processing. However, this study (Sharma and Dey, 2012) does not include any comparison to the other ML classifiers in order to measure the robust performance of proposed feature as well as the NNs classifier. Moreover, some of useful information and relations in the text might be missed during the selection process of the top n-ranked words. This study shows the visibility of using the NNs in our work. In our case, we will use all generated features with the NNs and compare its performance with the SVM.

Chen and Chiu (2009) proposed a method to classify the sentiment based on Neural Networks. The NNs was trained using three semantic orientation indexes: semantic orientation from association, Point-Wise Mutual Information (PMI) and Latent Semantic Analysis (LSI). An accuracy rate of 70% was achieved. They build first the bag-of-word feature model and then generated four sentiment orientation indexes depending on PMI or LSI. These values were then used to train the NNs. The classification was on one domain and polarity document level. Lastly, Dhande and Patnaik (2014) proposed an approach that merges NB classifier with the NNs classifier. They claimed that NB cannot capture the relationship between words due to the independence assumption that it is built on. This issue inspired them to combined the NNs with NB to capture the dependency that might be found between words. In their method, they reported increasing of the baseline NB model performance up to 81% in classifying the polarity of movie review document.

6.2. Methodology

This section illustrates our method to apply NNs to Arabic sentiment analysis. It starts with explaining the the NNs classifier structure. It then shows the details of how we use the NNs and compare it with another ML classifier with problem of Arabic sentiment analysis.

6.2.1. Neural Network Structure. The Neural Networks "NNs", sometimes is called Artificial Neural Networks "ANNs", is an information processing model that simulates in the way of a biological nervous systems. It is similar to how the brain manipulates information. The central element of this model is the structure of the processing system. It is composed of a vast number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example or training. The ANNs is configured for a particular application, such as linear/nonlinear regression, pattern recognition or data classification, through a learning process. The learning process in the biological systems involves adjustments to the synaptic connections that exist between the neurons (Bishop, 2006).Therefore, the NNs model also follows this updating to train the network by adjusting the weight between neurons.

The most common type of artificial neural networks consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units (Bishop, 2006). The input units represent the information that is fed into the network. The purpose of the hidden unit is determined by the the input units and the weights on the connections between the input and the hidden units. The number of units in the hidden layers can vary from no hidden layer (0: single neural) to n of hidden layers. Each layer has a set of weights associated with the inputs to the hidden layer. These layers apply a nonlinear function to the weighted sum of inputs. The first hidden layer

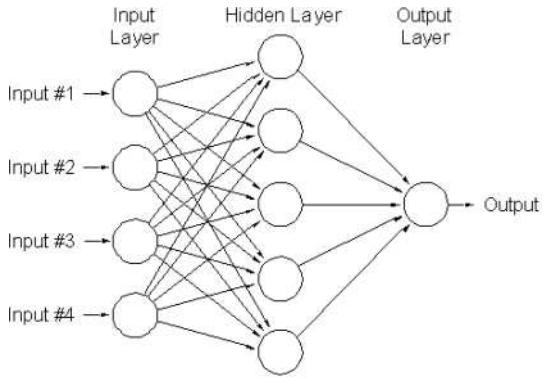


Figure 6.1: General overview of neural networks

operates on the input values while subsequent layers operated on the outputs of the previous hidden layer. In the end, the behavior of the output units depends on the behavior of the hidden units and the weights which is between the hidden and output units. The output layer produces a weighted sum of the outputs from the last hidden layer as the output of the neural networks. The neural networks learns the weights through an iterative process using training data. Figure 6.1 shows an example of a simple NNs structure.

The following equation computes the value of the predicted output Y for a given a set of X as inputs in a two layers, which are hidden and output layers, in a Neural Networks:

$$Y = \widetilde{h}(\widetilde{X}V)W,$$

where h is the activation function for the units in the hidden layer, it also represents as a nonlinear function used to transform the data. In addition, V and W are weights associated with the hidden and output layers. In general, Gradient descent is used to optimize the weights of the neural networks. Before using this method, the Log Likelihood of the data should be calculated for each value $g_{n,k}$, by this equation is:

$$LL(w) = \sum_{n=1}^{N} \sum_{k=1}^{K} t_{n,k} \log g_{n,k}$$

The Neural Networks model then learns by training with the values of the weights for the hidden "V" and output "W" layers. This process is done by maximizing the Log Likelihood between the values predicted "Y" and the target values "T" associated with the training data.

During the process of the training, two stages involve in the NNs. The first stage is called forward pass stage. In this pass, the input values are obtained and interact with the hidden layers and the output layer. The second phase is the backward pass. This step calculates the error that comes from the forward pass and adjusts the weights of the layers to minimize the error. This process will be propagating to all layers and units in the NNs. Those two stags should be repeated until the networks converge or it reaches the maximum number of iteration (Bishop, 2006). The training algorithm that used to make the NNs converge is Scale Conjugate Gradient descent (Møller, 1993).

6.2.2. Using Neural Networks with Arabic Sentiment Analysis. The same approach that is followed in Chapter 4 is used with the NNs classifier. At the first step, the text should be preprocessing in order to deliver it in a suitable format to the next stage. The second step generates the features that represent the text and produces the feature model that is explained in Chapter 4. The NNs classifier then is used to classify the sentiment in the Arabic text. Figure 6.2 illustrates the steps of our approach.

The NNs could be used to solve different machine learning problems such as prediction or classification. In our case, the NNs maintains the classification problems. The input layers should be equal to the number of features that we have in the feature model. For example,

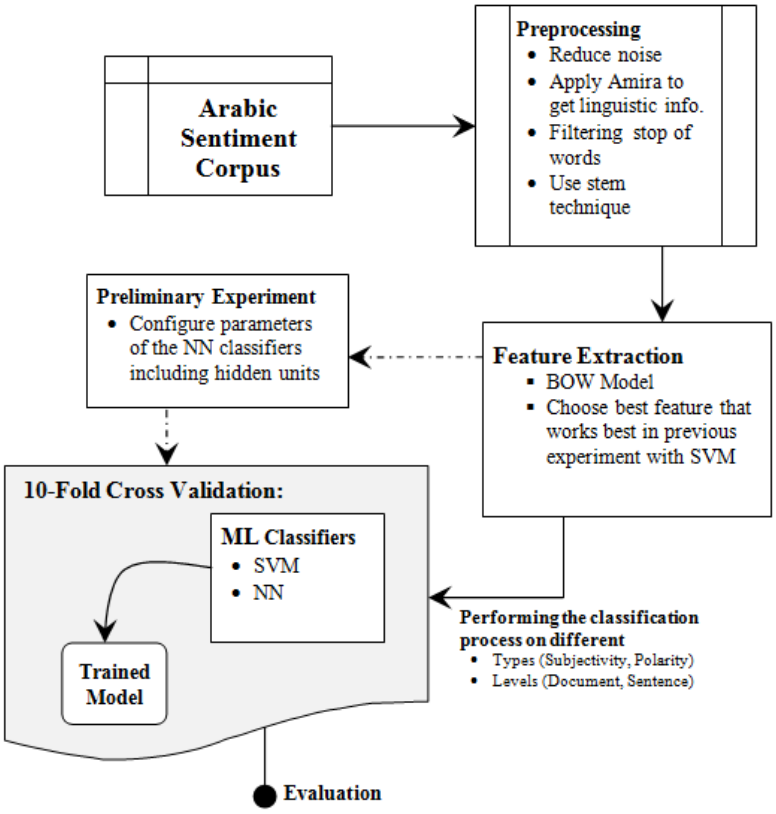


Figure 6.2: An overview of how the NNs compares with the SVM in Arabic sentiment analysis

suppose that we have a model of a particular domain. This model contains 2000 features which are the distinct words that represent that domain. As a result of that, the NNs must have 2000 units in the input layer.

The output layer should be equal to the number of categories in a particular problem. In our case, the output layer should contain two units in the case of subjectivity or binary polarity (polarity_2) classification. This layer is changed to have 3 units in the case of ternary polarity classification (polarity_3) when neutral class is included to the classification process. The classification result will be resolved by getting the maximum value of the unit. For example, suppose we have two units in the output layer, the NNs choose the first unit to represent the first category and the second one to the second class. The unit with highest value determines the class of the input to the NNs.

Choosing appropriate hidden layer is not a natural step. Every problem has different properties that have a role in selecting the number of units in the hidden layer (Bishop, 2006; Priddy and Keller, 2005). In addition, we could use multiple hidden layers with different units in each of them. This makes the process of choosing the optimal configuration of the hidden layer more complex. Our target with this work is to investigate the NNs with Arabic sentiment analysis and compare its performance with the linear SVM. Therefore, we rely on the basic and universal structure of the NNs with one hidden layer. Different configurations of the units are used in the hidden layer until we reach best performance of the NNs.

6.3. Experiments

This section explains the experiment setup that we follow during the evaluation process of the NNs and the SVM. The results are also discussed in the following sections.

6.3.1. Experiment Setup. The process of comparison is performed on different domains of sentiment datasets. It also includes different types of the Arabic language, MSA and DA. We use our sentiment corpus that we built in order to apply the NNs classifier for the problem of sentiment analysis in Arabic. The details of this corpus are explained on Chapter 3.

All tools that are used in Chapter 4 are also used to perform our experiment in this section. These tools include ARMIRA (Diab, 2009), and Scikit-learn (Pedregosa et al., 2011). The Scikit-learn has not implemented the NNs. Therefore, we use the Scikit-learn to prepare the feature model and performing the SVM classifier. The NNs package implementation was taken from Dr. Anderson Charles's website[1]. This implementation has the ability of using different hidden layers with different units for the NNs structure. The difference in this

[1]http://www.cs.colostate.edu/~anderson/cs545

package from the Scikit-learn is the way of how the data is processed, using either sparse or dense matrix. The Scikit-learn uses sparse matrix whiles the NNs package uses dense matrix. The feature model that is generated by the Scikit-learn is a sparse matrix. Therefore, the feature model is converted to the dense matrix before it feeds to the NNs classifier.

To evaluate these classifiers, we rely on calculating the F1-score by doing 5-fold cross validation. This evaluation also depends on the two other metrics that are precision and recall. In order to compute these metrics, the confusion matrix should be generated after the classification process. We follow the same method of evaluation metric that is illustrated in Chapter 4.

In our process of including the NNs as a classifier of the Sentiment of Arabic text, two analysis types are generated. The first type of experiment relates to measuring the process of the classification using each ML classifier over different types and levels. For example, the F1-score will be generated after the classification process for the SVM and the NNs in the case of subjectivity on document and sentence level.

The second type of experiments relates to measuring the time of the classification process. The time that is needed for each classifier to categorize a new text is calculated. In this experiment, the classifier is trained on some of the data and then performed the generated model on the testing part. We compute the time of performing the trained model of the classifier on the testing dataset part. This experiment is only carried out on sentence level classification and using only three different feature models.

Some studies say that the number of the units in the hidden layer is usually between the size of the input and size of the output layers (Priddy and Keller, 2005). This is not applicable in our case because we have a large deference between the input and output units.

For the time being, we trained the NNs with a certain range of hidden unit numbers and select the highest F1-score among them in the final experiment. Therefore, we follow these step to choose and determine the reasonable number of units in the hidden layer:

- A suitable number of candidate units for the hidden layer are determined: (5, 10, 15, 20, 25, 30, 35)
- Each of these candidate units, the NNs is trained and tested using 5-fold cross validation and calculated F1-score
- We choose the number of unit whose F1-score is highest

6.3.2. Feature Models. The main purpose of this chapter is to establish using the NNs with Arabic sentiment analysis and to compare its performance with the state-of-the-art ML classifier, the SVM. Due to the time consumption of the NNs training, not all of the features that are proposed or investigated in Chapter 4 are used. The primary target of the following experiment is to investigate the visibility of using the NNs to classify the sentiment in Arabic text. Therefore, only five feature models are chosen. The first one represents the baseline model that include the bag-of-word model. The second model represents one of the morphology features that we proposed. This feature is the one that captures the BPC of the sentence. The third one considers the polarity scoring of the text as a feature. The fourth feature is the stylistic feature that is either number of words or number of sentences. The last model contains the information about the negation in Arabic sentence. The details of the first four features are discussed in Chapter 4, while the last feature model is explained in Chapter 5. Table 6.1 displays these features.

Table 6.1: Different Type of features used with the NNs

Baseline	Includes all words in the model, except stop words
BPC	Captures the phrase chunk of the sentence
Polarity	Computes the polarity score of the text, using stem approach
Stylistic	Number of words, number of sentences
Negation	Include negation words with the model

6.4. Results

This section will discuss comparing the performance of two classifiers. The first section shows the accuracy of each classifier in different scenarios with five different feature configurations. The second section illustrates the performance of two classifiers in term of timing of the classification process.

Table 6.2 and Table 6.3 illustrate and compare the performance of the two classifiers. Table6.2 shows results at the sentence level classification whereas the document level is depicted in Table 6.3. The NNs refers to the Neural Networks, and the SVM represents the Support Vector Machine with linear kernel. The performance of the classification is measured using F1-score. The bold numbers in both the tables display the best results of the classifier within the particular feature model. For example, it seems the NNs achieves the best classification performance compared to the SVM with all feature models in case of subjectivity classification of news reviews domain, though we have insufficient samples to do a meaningful test for statistically-significant differences. The underlined values in the same tables indicates the best result of the classification process among different configurations.

For Example, the best outcome in subjectivity classification was with restaurant review domain achieved 73.2% that was carried out by using the SVM in polarity feature model.

It is clear from Table 6.2 that the performance of the NNs is better than the SVM in the case of subjectivity classification of the short sentence. This appears in news reviews and market reviews that contain short type of sentence. We also notice that the performance of subjectivity classification increases by more than 3% in the case of the news reviews. This may suggest that the NNs can capture the relationship and sentiment orientation in a short sentence better than the SVM. Moreover, it may indicate that the NNs might be able to capture the sentiment in the case of DA compared the MSA. This is because the news reviews and market reviews are written in dialect. In the case of polarity classification, the SVM plays the central role and achieves best result compared to the NNs, except in some instances with news domain. However, there is not a large gap between the results of using the NNs and the SVM.

In the instance of document level, Table 6.3 shows that the NNs outperforms the the SVM in the subjectivity classification. In the literature, the subjectivity classification is considered more difficult process than the polarity classification (Pang and Lee, 2008). This may show that the NNs classifier is able to perform better than the SVM and capture some relations in the text that help to increase the performance of the classification process. In the case of the polarity_2 classification, the SVM classifier achieves the best performance in most instances. However, the NNs increases the accuracy to 52.2% in the case of polarity_3 classification with movie reviews.

In general, the difference between the performances of the two classes is not remarkable in most cases. We cannot ensure that the one classifier works better than another classifier

Table 6.2: F1 score of the NNs versus. the SVM using different feature settings at the sentence level Classification

		Subjectivity		Polarity_2		Polarity_3	
		NNs	SVM	NNs	SVM	NNs	SVM
News Reviews	Baseline	**70.8%**	69.2%	57.6%	**58.1%**	56.6%	**57.3%**
	BPC	**70.8%**	69.0%	**57.2%**	56.9%	55.4%	**56.0%**
	Polarity	**72.8%**	70.1%	57.3%	**58.3%**	56.9%	**57.4%**
	No. of Words	**74.0%**	71.0%	55.9%	**57.0%**	54.5%	**56.0%**
	Negation	**71.6%**	71.2%	58.3%	**58.4%**	55.9%	**56.8%**
Restaurant Reviews	Baseline	69.3%	**71.0%**	82.0%	**83.4%**	70.9%	**73.2%**
	BPC	68.2%	**70.0%**	82.1%	**83.0%**	70.3%	**73.0%**
	Polarity	**72.3%**	71.3%	82.9%	**83.3%**	69.7%	**73.2%**
	No. of Words	71.4%	**72.0%**	81.9%	**82.0%**	69.9%	**73.0%**
	Negation	70.5%	**72.5%**	83.4%	**84.3%**	72.6%	**74.2%**
Market Reviews	Baseline	**90.1%**	89.3%	86.8%	**88.2%**	67.1%	**69.4%**
	BPC	**90.1%**	89.0%	88.7%	**90.0%**	67.5%	**69.0%**
	Polarity	**92.3%**	89.3%	87.0%	**88.1%**	68.0%	**68.9%**
	No. of Words	**90.1%**	86.0%	**88.0%**	**88.0%**	69.6%	**70.0%**
	Negation	**89.3%**	89.9%	89.3%	**90.4%**	69.8%	**70.3%**
Movie Reviews	Baseline	40.9%	**45.0%**	78.6%	**80.0%**	51.3%	**52.1%**
	BPC	40.1%	**44.0%**	79.1%	**81.9%**	**56.3%**	55.6%
	Polarity	41.7%	**44.8%**	79.1%	**80.1%**	52.4%	**52.8%**
	No. of Words	40.9%	**43.9%**	78.7%	**79.0%**	50.1%	**51.0%**
	Negation	40.2%	**46.4%**	81.0%	**88.7%**	50.8%	**51.7%**
News	Baseline	34.5%	**35.2%**	**80.4%**	80.1%	70.7%	**71.2%**
	BPC	31.9%	**33.9%**	**81.2%**	80.0%	**72.4%**	72.0%
	Polarity	34.6%	**36.3%**	**80.9%**	80.7%	**71.6%**	70.6%
	No. of Words	36.0%	**37.0%**	**79.8%**	78.9%	69.3%	**70.0%**
	Negation	35.3%	**35.9%**	**81.3%**	80.3%	**72.0%**	71.8%

in a particular case. However, this suggests that the NNs may play some role in the future for Arabic sentiment analysis. It could be used in the case of DA due to its performance with the short reviews, or with the three types of polarity when the neutral class is added. The NNs is able where the SVM fails in this, perhaps due to the non-linearity in the data that comes from the nature of the Arabic language. In some cases, the time is an important factor in the analysis process. The next results will compare the times of classification for both classifiers.

Table 6.3: F1 score of the NNs versus. the SVM using different feature settings at the document Level Classification

		Subjectivity		Polarity_2		Polarity_3	
		NNs	SVM	NNs	SVM	NNs	SVM
News Reviews	BOW	**88.4%**	88.1%	56.1%	**56.4%**	**58.6%**	58.1%
	BPC	**89.2%**	89.0%	56.3%	**57.0%**	**58.0%**	**58.0%**
	Polarity	**95.7%**	95.2%	54.8%	**55.7%**	52.6%	**57.1%**
	No. of Sentences	**88.4%**	88.0%	56.4%	**57.0%**	**59.2%**	58.0%
	Negation	**88.8%**	88.3%	56.7%	**56.9%**	**58.6%**	57.7%
Restaurant Reviews	BOW	**96.5%**	96.2%	**86.0%**	85.3%	**68.2%**	67.0%
	BPC	**96.5%**	96.0%	**86.4%**	86.0%	**67.5%**	67.0%
	Polarity	**87.8%**	87.5%	**85.1%**	84.8%	63.5%	**67.0%**
	No. of Sentences	**96.4%**	96.0%	**85.6%**	85.0%	**68.6%**	67.0%
	Negation	**96.0%**	**96.0%**	**87.1%**	86.2%	**68.7%**	67.6%
Market Reviews	BOW	**93.9%**	93.4%	88.3%	**90.0%**	66.2%	**70.0%**
	BPC	**93.7%**	93.0%	86.6%	**90.0%**	67.0%	**70.9%**
	Polarity	**94.2%**	93.1%	88.7%	**89.9%**	68.9%	**70.2%**
	No. of Sentences	**93.8%**	93.0%	86.5%	**89.0%**	66.4%	**70.0%**
	Negation	**94.1%**	93.4%	90.6%	**91.7%**	69.8%	**71.2%**
Movie Reviews	BOW	NA	NA	**81.0%**	80.0%	47.1%	44.5%
	BPC	NA	NA	80.2%	**81.0%**	50.6%	50.2%
	Polarity	NA	NA	78.9%	**80.1%**	42.0%	43.0%
	No. of Sentences	NA	NA	71.4%	**80.0%**	42.8%	42.6%
	Negation	NA	NA	80.3%	**82.7%**	52.2%	51.9%
News	BOW	**63.5%**	63.4%	**77.2%**	76.4%	**65.3%**	**65.3%**
	BPC	63.4%	**63.9%**	77.8%	**77.9%**	**65.5%**	64.0%
	Polarity	**66.0%**	62.4%	**76.5%**	75.7%	53.9%	**64.2%**
	No. of Sentences	**65.5%**	65.0%	**78.5%**	76.9%	65.0%	**65.7%**
	Negation	**63.5%**	61.5%	**78.5%**	77.3%	**65.9%**	64.9%

Table 6.4 displays the result of comparing the classification time of each classifier. For the sake of conciseness, one classification level, sentence level, is used in this experiment. This result is only performed on three different features models. These models are bag-of-word (baseline), Base Phrase Chunk (BPC), and negation feature to the model. The numbers are in Table 6.4 shows the time that the classifier needs to perform the classification on the testing dataset. The results are shown in milliseconds and represent the average of the performing the classifier on different dataset domains. The machine that is used to perform this experiment has 4 Gigabyte RAM and CPU Intel Core Quad with 2.83 GHz.

Table 6.4: An Average of classification time (testing time) of the NNs and the SVM

	Subjectivity		Polarity_2		Polarity_3	
	NNs	SVM	NNs	SVM	NNs	SVM
BOW	0.0102	0.6042	0.0036	0.131	0.0046	0.2222
BPC	0.0098	0.6142	0.0038	0.1364	0.005	0.2292
Negation	0.0094	0.6288	0.004	0.1344	0.0046	0.2306

It is noticeable that the NNs classifier does not need as much time as the SVM needs. For example, the NNs needs around 0.01 millisecond to perform the classification on the testing data using baseline model. That means the SVM needs 60 times more the time that the NNs needs in the case of subjectivity classification. The scenario is the same in case of polarity classification. This difference comes from the difference in the nature of the two classifiers. The SVM needs to build hyperplanes between classes and each of them contains many support vectors, as explained in Chapter 4. For example, the SVM will have more than 4000 support vectors in the case of subjectivity classification. On the other hand, we know that the NNs is built as the connected network of three different layers, as explained earlier in this chapter. In our experiment, we trained the NNs with 25 units in the hidden layer. This makes the classification process (applying the trained model on a new data) much faster than the SVM that needs to calculate the value of all the support vectors. However, there is some drawback of using the NNs over the SVM that relates to the training time of the NNs. The training time of the NNs is much higher compared to the SVM. For example, the NNs needs 35 milliseconds to trained, whereas the SVM takes around 0.11 milliseconds. Finally, there is no optimal solution to any problem but the problem's environment controls the method or the approach that is prepared.

6.5. Chapter Summary

This chapter introduces using the NNs classifier with Arabic sentiment analysis. The popular the NNs structure is used to resolve the classification process of sentiment in Arabic text. The structure of the NNs contains input, one hidden layer and an output layer. The input layer consists of as many units as the number of features in the feature model. The hidden layer contains 25 units after performing some initial experiments to choose the reasonable number of units in this layer. The output layer would provide two or three units depending on the categories in the classification problem.

The performance of using the NNs in this problem has been compared with the state-of-the-art ML classifier, the SVM. The comparison includes measuring the performance of the classification process in terms of accuracy and timing. Due to the space and time limitation, this evaluation used only five different feature models. The first part of the evaluation includes computing the F-1 score of the classification process. The second part compares the time that is needed for each classifier in order to perform the classification problem.

In most cases, the SVM outperforms the NNs classifier. However, the NNs classifier can achieve a better result in the case of the the Dialect Arabic. It also increases the accuracy in the case of polarity_3 classification when the neutral category is added. In general, the difference in the performance between the two classifiers is not significant. This might suggest that the two classifiers have an equivalent performance in the case of Arabic sentiment analysis. However, the NNs outperforms the SVM in term of the timing. The SVM needs around more than 30 times the amount of time that the NNs needs to perform the classification process with new data. Therefore, the NNs classifier is preferable in the case of a sensitive system that needs quick answers about the sentiment in a text. The

big difference in the timing comes from the difference in the nature and structure of each classifier. However, the NNs need more time to be trained compared to the SVM classifier.

At document level (Subjectivity) classification, it seems that the NNs able to capture some non-linearity that the SVM cannot. This is the same case in short sentence classification. The NNs also work in the case of polarity_2 with long sentence classification. The NNs works in restaurant and news domains in polarity2 classification, as well as in movie reviews with polarity_3.

This chapter highlights the using of popular ML classifier, the NNs, in the field of sentiment analysis of Arabic language. The results are promising to continue using the NNs to classify the sentiment in the Arabic language. The only issue with the NNs classifier is the training time that is needed. However, this issue might be resolved by the advancement in the present or future machine speed. In addition, this is not the only way to improve Arabic sentiment analysis. Some other improvements should be made in a different direction. One of these directions is to learn knowledge of the Arabic sentiment from one domain and apply that to another domain. The next chapter illustrates this approach that may help in the field of sentiment classification of the limited resource language such as Arabic.

CHAPTER 7

Arabic sentiment Analysis Across Domains

In the previous chapters, the work focuses on analyzing the sentiment analysis in Arabic text by introducing different features and using different ML classifiers. All of these proposed methods were carried out within each specific domain. For each proposed feature the ML classifier is trained and tested on the same domain of the dataset such as newswire. This help the classifier to learn sentimental knowledge from a specific domain, but could the learned knowledge be applied to another domain such as movie review? This might help in case of the limited sentiment corpora in different domains. The limitation of these resources are common in the Arabic language. Even though there is a few of an Arabic sentiment corpora, these corpora dedicates to the one type of Arabic language, MSA. Therefore, applying learned knowledge from one domain to another one might be useful in the case of Arabic sentiment analysis. This chapter will focus on applying Arabic sentiment analysis using the cross domain technique.

This chapter explains some introductory information about the cross-domain method in the first section. The second section illustrates our proposed methodology of applying the cross-domain method to Arabic sentiment analysis. The experiments and the results are discussed in Section 3. The last section summarizes this chapter.

7.1. Introduction

The sentimental feeling is sometimes expressed differently from one domain to another domain. This differentiation between domains is costly because it requires annotated data for each new domain before building sentiment analysis system. Therefore, one of the possible solutions is to perform cross domain method on sentiment analysis.

In the English language, the idea of cross domain sentiment analysis has been investigated. Aue and Gabon (2005) tried to compare the results of using four different training and testing domains. Different n-gram models are used including uni-gram, tri-gram, and n-gram feature sets. They also pay attention to negation in their work. Their investigation shows a high performance for in-domain sentiment classification using the SVM classifiers. On the other hand, the results were mixed for cross-domain sentiment classification, ranging from barely above chance to near the accuracy of in-domain .

Glorot, et al.,(2011) found an approach to using current label data on specific domains to be used in different domains by generalizing the feature vector that works best in both domains. They proposed deep learning method that helps to extract a meaningful representation for each review. The ML classifier was then trained with these high level feature representations. They claimed that their approach outperformed the state-of-the-art methods on a benchmark composed of reviews of 4 types of Amazon products.

Bollegala, et al.,(2011) developed a method based on a sentiment sensitive thesaurus (SST). They used this method for performing cross-domain sentiment analysis. The SST was constructed by using labeled data from multiple source domains, and unlabeled data from source and target domains and computing the relatedness of features. This method may help to handle the mismatching between features in cross domain processing. This SST is used to expand feature vector during the train and test the binary classifier.

To the best of the author's knowledge, no work has been done with cross domain in Arabic sentiment analysis. Therefore, the work in this chapter would be a corner stone of Arabic sentiment analysis based on cross domain concept. This work also may help to improve the sentiment classification of Arabic text especially with the DA that does not have a special

basic natural language processing tool such as morphology analyzer. In addition, it may help and encourage using the existent labeled Arabic sentiment data with a new domain that has not been labeled yet.

To sum up, the machine learning sentiment classification techniques require large amounts of labeled training data. Building these labeled corpora consumes time and expenses. In this chapter, we explore various cross domain strategies for training classifiers in Arabic sentiment field. We present different strategies to customize sentiment classifiers to a new domain in the absence of labeled data in that domain. The next section will describes these strategies.

7.2. Methodology

In the previous works that have been done on sentiment classification, the classifier is trained using labeled data and then it is applied and tested on the same domain. This process is called single-domain, or in-domain sentiment classification (Aue and Gamon, 2005). With this approach, the classifier will learn the knowledge from the features that are chosen from a particular domain. It then uses this knowledge on new data from same domain. On the other hand, the cross-domain aspect is a method when classifier is trained on a specific domain and then applied on another different domain.

With applying cross-domain method on Arabic, we can find whether some of the knowledge could be transferred from one domain to another. In addition, the classifier may learn the sentiment orientation in MSA "Modern Standard Arabic" domain and use it with a DA "Dialect Arabic" domain. Therefore, this will help to save more time in labeling a new data with a particular type of DA.

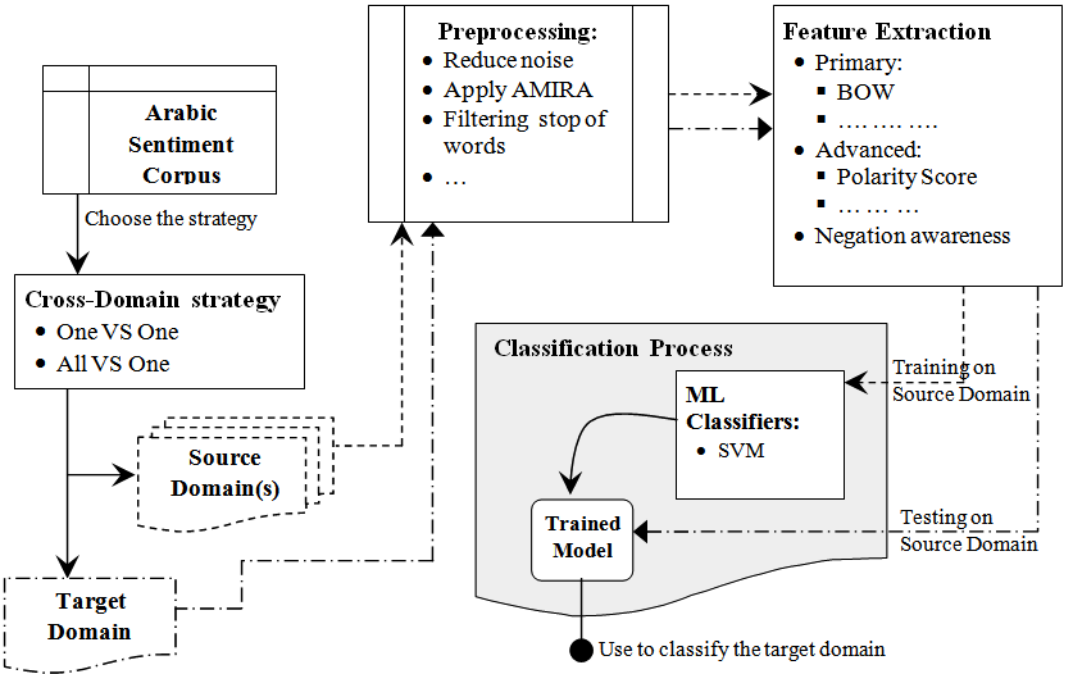


Figure 7.1: A method of performing cross-domain Arabic sentiment analysis

Figure 7.1 illustrates the method that we apply to perform the sentiment analysis in Arabic text using the cross domain method. There are two types of data in the process of classification, these are the source and the target data. The source data is the dataset domain that the classifier receives the knowledge from and trained on. The target data is the dataset domain which the learned knowledge from the source domain is applied on. The first step is preprocessing the source data to build suitable feature vector model, as explained in Chapter 4. The second phase uses all data in the source domain to train the classifier. The generated model after the training phase is used on the target domain.

Two strategies are proposed in this chapter in order to perform the cross-domain method on Arabic sentiment analysis. The first one is All-to-One, whereas the second is One-to-One. In the first method, all dataset domains will be chosen as the training data except one domain that is used for testing. In the One-to-One approach, one particular domain is used

as training and another different domain is used for testing. By performing these methods, different knowledge could be learned by the classifier. In addition, the investigation about the specific domains that might work well with other different domains may be found.

7.2.1. One-to-One Across Domain. This technique involves examining the cross-domain concept between one specific domain to another single domain. This strategy would find whether the knowledge of one domain such as, movie reviews might be transfer to another type of domain, such as newswire. In addition, they would prove whether the sentiment orientation of the words could be preserved across different types of the Arabic language. This could be performed by training the classifier on one MSA domain and testing on DA domain.

In this section, each of the domains would be chosen as a source dataset to train the classifier and then to test the generated model on each of other domains. Suppose that we start with market reviews as a source dataset, the classifier then will be trained on this data alone. The next step will test each of the other domains on the generated model. The performance of each of them is compared with the baseline model of in-domain classification process for each of the data. The following steps describe the details of this experiment:

- Lets:
  - D is the set of data of different domains
  - TD is the Target Domain, and SD is the Source Domain
  - $Model_{cross}$ is the model generated by classifier using cross-domain method
  - $F_{1-cross}$is the metric value that is calculated using cross-domain, $F_{1-within}$ is computed within the same domain
- for $d_i \in D$

- $DS = d_i$
- Train the ML Classifier "SVM" with DS for particular classification process "$Model_{cross}$" , such as sentence level subjectivity classification
- for $d_j \in D$, $where\, d_j \neq d_i$
  * $DT = d_j$
  * Test the $Model_{cross}$ with DT and record the $F_{1-cross}$
  * Train and test ML Classifier "SVM" with DT and record the $F_{1-within}$

• Evaluate the recorded values of $F_{1-cross}$ with $F_{1-within}$

7.2.2. All-to-One Cross Domain. The main aim in this technique is to find whether the expanding of trained domain would help the Arabic sentiment classification process. This would inform us if the classifier can learn more knowledge from the domain extension. In addition, this mechanism may increase the performance of the classification of the Dialect Arabic domains when there is lack of resources and tools in the field of NLP.

This approach will group all domains in one set as a training dataset except for one domain. For example, lets say that the Market reviews would be the target dataset "testing dataset", then all other domains should be grouped as one dataset called "source dataset" domain. The feature model then is built using the baseline model that includes the negation words as vocabulary dictionary. The SVM classifier is trained using the source dataset. The target dataset is used for the testing the generated model.

To evaluate the method, the performance of ML classifier should be compare with the baseline model of performing the classifier on a particular domain. For example, lets assume that the target domain is Movie reviews, then the source would be the other domains together as one. The performance of the "cross-domain" of using movie review as a target dataset

would be compared to the performance of the classifier using movie review for training and testing phase which is called "in-domain" classification. The following steps describe the details of this experiment:

- Lets:
  - D is the set of data of different domains
  - TD is the Target Domain, and SD is the Source Domain
  - $Model_{cross}$ is the model generated by classifier using cross-domain method
  - $F_{1-cross}$is the metric value that is calculated using cross-domain, $F_{1-within}$ is computed within the same domain
- for $d_i \in D$
  - $DT = d_i$
  - Groups all $d_j \in D$, $where\, d_j \neq d_i$and makes Them in DS
  - Train the ML Classifier "SVM" with DS for particular classification process "$Model_{cross}$" , such as sentence level subjectivity classification
  - Test the $Model_{cross}$ with DT and record the $F_{1-cross}$
  - Train and test ML Classifier "SVM" with DT and record the $F_{1-within}$
- Evaluate the recorded values of $F_{1-cross}$ with $F_{1-within}$

7.3. Experiment

This section describes the details of the experiments that are carried out in this chapter. The first section illustrates the setup of these experiments. The second section shows the details of using the cross-domain method with Arabic sentiment and discusses their results.

7.3.1. Experiment setup. Different of experiments were undertaken using Support Vector Machine classifier (SVM) with linear kernel, to evaluate the cross-domain concept in sentiment analysis for Arabic text, . As a basic step, the uni-gram model is applied for the learning and testing process. We relied on the scikit-learn library (Pedregosa et al., 2011) for using machine-learning classifiers. The classification process is achieved on both classification levels, sentence and document. It is also performed on the different classification types which are subjectivity, binary polarity and ternary polarity classification. In the cross-domain classification process, the SVM is trained on the source dataset domain and then tested on the target dataset domain. In the in-domain classification process, the 5-fold cross validation is used, as explained in Chapter 4. The F1-score was calculated , to evaluate these classifiers.

7.4. Results

7.4.1. One-to-One Cross Domain. Table 7.1 and 7.2 displays and compares the results of performing cross-domain and in-domain for sentiment analysis for the Arabic language. Table 7.1 shows the results at the sentence level classification and Table 7.2 illustrates the document level classification. Each main row in these tables represent the results of training the classifier on one dataset domain and then testing on each individual domains. The first column shows the source domain dataset that used for training the classifier. The second column illustrates the target domains that used during testing phase of the classifier. The next columns display the result of each classification types. For each type of classification both methods, cross-domain "$F_{1-cross}$" and in-domain "$F_{1-within}$" are compared. For example, Table 7.1shows the results of the training classifier on news reviews (as training dataset) and testing the other domains on the generated model. In subjectivity, the classifier achieves

Table 7.1: Comparing one-vs-one cross-domain technique with in-domain at the sentence level of Arabic sentiment classification

		Subjectivity		Polarity 2		Polarity 3	
Training On	Testing On	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$
News Reviews	Restaurant Reviews	57.5%	**71.0%**	43.0%	**83.4%**	37.7%	**73.2%**
	Market Reviews	55.0%	**89.3%**	68.1%	**88.2%**	42.1%	**69.4%**
	Movie Reviews	**57.6%**	45.0%	51.9%	**80.0%**	35.0%	**52.1%**
	News	**45.6%**	35.2%	54.1%	**80.1%**	40.9%	**71.2%**
Restaurant Reviews	News Reviews	46.9%	**69.2%**	47.8%	**58.1%**	36.2%	**57.3%**
	Market Reviews	58.6%	**89.3%**	23.2%	**88.2%**	14.0%	**69.4%**
	Movie Reviews	**60.6%**	45.0%	65.1%	**80.0%**	48.8%	**52.1%**
	News	**51.3%**	35.2%	47.2%	**80.1%**	47.6%	**71.2%**
Market Reviews	News Reviews	54.7%	**69.2%**	53.9%	**83.4%**	37.1%	**57.3%**
	Restaurant Reviews	61.5%	**71.0%**	31.5%	**71.0%**	29.6%	**73.2%**
	Movie Reviews	**57.8%**	45.0%	38.7%	**80.0%**	33.6%	**52.1%**
	News	**42.6%**	35.2%	50.7%	**80.1%**	40.1%	**71.2%**
Movie Reviews	News Reviews	36.4%	**69.2%**	51.9%	**83.4%**	29.5%	**57.3%**
	Restaurant Reviews	47.0%	**71.0%**	58.1%	**88.2%**	51.6%	**73.2%**
	Market Reviews	31.9%	**89.3%**	16.8%	**89.3%**	19.4%	**69.4%**
	News	**38.4%**	35.2%	34.7%	**80.1%**	50.2%	**71.2%**
News	News Reviews	32.6%	**69.2%**	58.5%	**83.4%**	36.9%	**57.3%**
	Restaurant Reviews	32.7%	**71.0%**	50.2%	**71.0%**	40.0%	**73.2%**
	Market Reviews	7.0%	**89.3%**	51.5%	**88.2%**	10.6%	**52.1%**
	Movie Reviews	**61.2%**	47.0%	45.9%	**80.0%**	39.4%	**47.0%**

89% using in-domain "$F_{1-within}$" approach and 55% in cross-domain "$F_{1-cross}$" method with market reviews as target dataset (testing dataset) and news reviews as the source dataset. The symbol NA in Table 7.2 refers to a not applicable results for the classification process because the movie reviews domain does not have any subjective document. The boldfaced value shows the best performance achieved across different methods.

It is clear from these tables that the cross-domain does not outperform the in-domain approach in Arabic sentiment analysis on all types and levels of classification. This comes from the nature of the sentiment analysis problem. It also suggests that Arabic sentiment analysis is domain dependent. We notice that there is a huge difference between the results of the in-domain "$F_{1-within}$" and the cross-domain "$F_{1-cross}$" in most cases of the classification

Table 7.2: Comparing one-vs-one cross-domain technique with single-domain at the document level of Arabic sentiment classification

		Subjectivity		Polarity 2		Polarity 3	
Training On	Testing On	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$
News Reviews	Restaurant Reviews	88.8%	**96.2%**	36.3%	**85.3%**	25.0%	**67.0%**
	Market Reviews	86.7%	**93.4%**	41.5%	**90.0%**	20.9%	**70.0%**
	Movie Reviews	NA	NA	41.3%	**80.0%**	30.0%	**44.5%**
	News	52.1%	**63.4%**	49.7%	**76.4%**	42.9%	**65.3%**
Restaurant Reviews	News Reviews	68.3%	**88.1%**	**60.2%**	56.4%	47.6%	**58.1%**
	Market Reviews	86.2%	**93.4%**	63.9%	**90.0%**	43.4%	**70.0%**
	Movie Reviews	NA	NA	37.9%	**80.0%**	30.6%	**44.5%**
	News	52.5%	**63.4%**	56.1%	**76.4%**	44.9%	**65.3%**
Market Reviews	News Reviews	64.3%	**88.1%**	**63.6%**	56.4%	51.5%	**58.1%**
	Restaurant Reviews	86.1%	**96.2%**	69.9%	**85.3%**	52.5%	**67.0%**
	Movie Reviews	NA	NA	18.9%	**80.0%**	11.3%	**44.5%**
	News	52.9%	**63.4%**	42.2%	**76.4%**	36.4%	**65.3%**
Movie Reviews	News Reviews	NA	NA	26.8%	**56.4%**	21.4%	**58.1%**
	Restaurant Reviews	NA	NA	52.7%	**85.3%**	41.9%	**67.0%**
	Market Reviews	NA	NA	37.9%	**90.0%**	26.4%	**70.0%**
	News	NA	NA	52.0%	**76.4%**	46.6%	**65.3%**
News	News Reviews	55.2%	**88.1%**	27.3%	**56.4%**	21.8%	**58.1%**
	Restaurant Reviews	83.0%	**96.2%**	55.4%	**85.3%**	44.3%	**67.0%**
	Market Reviews	84.7%	**93.4%**	37.8%	**90.0%**	26.4%	**70.0%**
	Movie Reviews	NA	NA	47.7%	**80.0%**	33.8%	**44.5%**

process. However, there are some promising results using the cross-domain approach. The first noticeable point is the increasing of the result in the case of polarity_2 of document level classification with news reviews when the classifier is trained on restaurant reviews or market reviews, Table 7.2. The improvement may occur because of the nature of those domains. They are in different fields but all of them are user's feedback. This may indicate that the user's feedback about something may have the same nature of sentiment in the Arabic language. The other reason may come from the training data. The restaurant or the market reviews has concise and clear sentimental orientation words and styles that express the opinion clearly compared to the news reviews. Therefore, using these domains give the

classifier good knowledge about the user review in general and helps more in the case of news reviews domain, Table 7.2 , in polarity_2.

In the Sentence-Level classification, Table 7.1, the result is the same as the document level classification but with other place of improvement. This improvement includes the long sentence domains which are the newswire and movie reviews. We notice that the performance of cross-domain in the case of subjectivity classification is better than the in-domain method when the classifier is trained on medium or short sentence. This improvement may come from the nature of these domains. In the newswire and movie review, there are many objective sentences that may make the process of the subjectivity more complex. Therefore, the classifier may gain some extra knowledge by adding these objective sentences from that domain. By training the classifier on some domain that contain precise and clear subjective sentences may help to classify other domains when the objective sentences are the dominant ones. In the case of sentence polarity classification, the cross-domain does not add any extra knowledge to the classifier and even hurt the classification process compared to the in-domain.

7.4.2. All-to-One Cross Domain. Table 7.3 and Table 7.4 represent the results of the cross-domain classification process using the All-to-One method. The performance of cross-domain is compared with the in-domain in each classification types. For each domain dataset, the classifier will be trained on all domains except the particular testing domain. The generated model would be used on the testing domain. For example, the market reviews is considered as the testing dataset "target domain", so the classifier should be trained on all other domains as one except the market reviews. The results of cross-domain for each classification types, such as subjectivity, will be recorded as well as the single domain

classification process when the training and testing are performed on the same domain with 5-fold cross validation. The boldfaced numbers in both tables represents the best result achieved across the two methods that are cross-domain and in-domain using All-to-One technique.

Both tables illustrate that the cross-domain classification process does not outperform the in-domain classification in the most cases. The expanding the source domain with extra data does not help at all. This suggests that the Arabic sentiment problem is domain dependent. That means when the classification process is performed on a particular domain the output of that process might not work with different domains. Therefore, the new domain needs to be manipulated in a native way and build the process of classification from the scratch. However, there is some promising results with using the All-to-One cross-domain method.

The result of the classifier increases by more than 15% in the case of subjectivity sentence level classification with movie reviews and newswire, Table 7.3. The subjectivity classification is considered more difficult than the polarity classification because there are more objective sentences the appear during the process of the classification (Pang et al., 2002). When the classifier is trained on the domain such as newswire, this domain has a lot of objective sentences that may confuse the classifier while it learns about the subjectivity. On the other hand, the classifier could learn more about the subjectivity from the domain where there are comparable number of objective sentences to the subjective sentences. For example, the performance of cross-domain increases the subjectivity classification to 72.3% from 35% in the news domain, Table 7.3.

These improvements in these domains were recorded also in the previous method, one-to-one cross-domain. The best improvement that is recorded in Table 7.1 is around 14.2%

Table 7.3: All-vs-One Cross-Domain and In-Domain at the sentence level classification

	Subjectivity		Polarity 2		Polarity 3	
	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$
News Review	48.0%	**69.2%**	52.5%	**58.1%**	38.4%	**57.3%**
Restaurant Reviews	62.5%	**71.0%**	39.4%	**83.4%**	38.3%	**73.2%**
Market Reviews	62.0%	**89.3%**	32.7%	**88.2%**	22.3%	**69.4%**
Movie Reviews	**63.4%**	45.0%	54.0%	**80.0%**	46.3%	**52.1%**
News	**72.3%**	35.2%	48.5%	**80.1%**	47.7%	**71.2%**

when classifier trained on newswire domain and tested on movie reviews, and 16.1% when newswire used during testing and restautant reviews for training. In this method, when the source dataset is expanded, the improvement is around 16.7% with movie reviews, and 36.9% with news. This suggests that the expanding the training set may improve the accuracy of the classifier on the target dataset.

In document classification, Table 7.4, the cross-domain method behaves well on one case with the news reviews domain in polarity_2 classification. This also reveals the same improvement was recorder in the case of one-to-one method when the classifier is trained on either market or restaurant reviews. The average improvement in one-vs-one method was around 5.3%, but with the All-to-One, the improvement is around 6.2%. This may suggest that the expanding source domain of the training dataset might improve the result of classification on the target source. In addition, the nature of market and restaurant reviews has more clear and precise sentiment sentences compared to the News review which may lead to the improvement in the results.

7.4.2.1. *Modern Stranded Arabic Versus Dialect.* Regarding the types of Arabic language, this method also has similar performance to the one-to-one method. The restaurant review has mixed types of MSA and DA, adding this domain in the source dataset or using it alone does not help much in both cases, either in DA or MSA domain. Therefore, we believe

Table 7.4: All-vs-One Cross-Domain and In-Domain at the document level classification

	Subjectivity		Polarity 2		Polarity 3	
	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$
News Review	66.7%	**88.1%**	**63.9%**	56.4%	50.3%	**58.1%**
Restaurant Reviews	87.4%	**96.2%**	71.5%	**85.3%**	56.1%	**67.0%**
Market Reviews	86.5%	**93.4%**	60.1%	**90.0%**	42.5%	**70.0%**
Movie Reviews	NA	NA	35.7%	**80.0%**	22.6%	**44.5%**
News	51.5%	**63.4%**	57.1%	**76.4%**	52.1%	**65.3%**

Table 7.5: Result of cross-domain using DA-vs-MSA at sentence level classification

		Subjectivity		Polarity 2		Polarity 3	
Source domains	Target domains	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$
DA	MSA	**65.3%**	44.3%	53.0%	**64.9%**	45.0%	**61.0%**
MSA	DA	40.4%	**73.5%**	50.1%	**66.2%**	38.5%	**63.0%**

that the type of Arabic language plays a role in sentiment analysis, so the classifier should be trained on the same type of Arabic language in order to get more accurate results.

More experimentation is performed in this area to find the actual effect of Arabic language types with cross-domain methods. Table 7.5 and Table7.6 illustrates the results of performing DA on MSA domains and vice verse. The first row represents DA domains as a source dataset "training set" and MSA domains as a target dataset "testing set". News reviews, Market reviews, and Restaurant reviews shows the DA type of Arabic. The other two domains, news texts and movie reviews show MSA domains because all their sentences are written in MSA. In the in-domain classification, the target domain, either DA or MSA domains, is used during the training and the testing of the classifier. The classification is carried out on different levels and types as is shown in Table 7.5 and Table 7.6. The boldfaced values display the best performance achieved across different methods.

Both tables illustrate that the in-domain classification process outperforms the cross-domain method. This suggests that Arabic sentiment analysis is a dialect language dependent

Table 7.6: Result of cross-domain using DA-vs-MSA at document level classification

		Subjectivity		Polarity 2		Polarity 3	
Source domains	Target domains	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$	$F_{1-cross}$	$F_{1-within}$
DA	MSA	NA	NA	50.7%	**78.9%**	42.4%	**57.4%**
MSA	DA	75.5%	**91.6%**	44.4%	**82.3%**	34.1%	**65.8%**

problem. In the case of sentence level classification, Table 7.5 shows that the cross-domain outperforms the in-domain method in one case, at sentence level subjectivity classification. This reveals the same behavior of the one-to-one or all-to-one approaches. The domains that are used in the DA side are considered short to medium sentence type with clear sentiment orientation. The target domain has long sentence type with more objective sentences. These factors help to improve the classification in the case of using cross-domain instead of using in-domain. This improvement cannot prove that the DA domain could be used to investigate the sentiment of MSA. The other results shows that the sentimental words orientation differs from DA to MSA. Therefore, the classifier is needed to be trained on the same type of the Arabic language in order to gain higher accuracy rates.

7.5. Chapter Summary

In Arabic sentiment analysis, when the lack of the specialized sentiment corpora appear in some language, there is a limited amount of research that applies cross-domain approaches. This concept of using cross-domain is introduced in this chapter. The cross-domain is the method to learn a model from particular labeled domains and then applying this knowledge to another unlabeled domain. This method saves time and effort needed to label the new domain of dataset. In the case of Arabic, there are limited sentiment corpus in MSA,

moreover, nothing in DA. Therefore, this work starts to build the first step toward applying cross-domain in Arabic sentiment analysis.

There are two cross-domain methods evaluated to work with Arabic sentiment analysis. The first approach is the one-to-one croos domain. In this method, the ML classifier tries to learn sentiment model from one particular domain and applies that model to another domain. The second mechanism is the all-to-one. In this approach, the classifier builds sentiment knowledge from different domains and applies these experiences to one different domain.

The experiments show that the in-domain outperforms the cross-domain methods. The low performance of the cross-domain comes from the nature of the Arabic sentiment problem. Arabic sentiment analysis seems to be domain dependent. This indicates that the ML classifier should be trained on the same domain that needs sentiment analysis system. However, there are some promising results that may encourage continuing in this direction. The cross-domain improves the result of subjectivity at the sentence level classification with the long sentence type domain. This reveals that the cross-domain method helps the classifier to learn exact sentiment orientation from short or medium domains that have clear sentiment sentences and is able to apply this model on long domain that have more objective sentences. In addition, this method plays some role in improving the polarity classification with learning from the domains that have clear sentiment orientation in the text and applying them to other domains.

This chapter produces some general conclusions. Arabic sentiment analysis should be considered dialect domain dependent. Our method that was applied with the cross-domain classification does not consider any type of adaptation between different domains. However,

the subjectivity classification results improve using cross-domain when it trains on medium or short sentence type and is applied on long text. This may encourage the new researcher to focus more in this field. Adding some adaptation mechanisms between the domains or making generalizations of the obtained features to work well in the target domain may help to increase the accuracy of cross-domain method.

In science, at no point is an optimal or complete level reached. There is always another level to take research to.The next chapter will give summary of this dissertation and some possible future directions to Arabic sentiment analysis.

CHAPTER 8

Conclusion and Future Work

This dissertation addresses the task of document- and sentence-level sentiment analysis in Arabic text. This task is considered a central part of other NLP tasks, such as question answering systems (Pang and Lee, 2008). In addition, the sentiment analysis field plays a primary role in different applications such as predicting sales performance (Liu et al., 2007), using reviews to rank products and merchants (McGlohon et al., 2010), or predicting the election results by analyzing Twitter (Tumasjan et al., 2010). This task has been carried out in multiple languages, but mostly in English, whereas there is a limited amount of work in the case of the Arabic language. Arabic is considered a rich morphology language that may need new approaches to achieve the sentiment classification tasks. These limitations, characteristics and challenges of Arabic lead to the need for new Arabic sentiment resources, proposing and studies involving a new features in a suitable manner, comparisons of ML classifiers, and dealing with the central sentiment influential factor, negation. The main research questions that are addressed in this dissertation will be revisited in this chapter:

- Are there enough sentiment corpora for the Arabic language?
- How should a highly inflectional and morphological language such as Arabic be treated in sentiment analysis?
- What is the effect of negation in Arabic sentiment analysis?

The following sections will show the main findings of this dissertation. The next section explains the contributions of our work. Possible future work paths will be illustrated in the last part of this chapter.

8.1. Main Findings

The dissertation addresses the task of examining document- and sentence-level sentiment classification of Arabic text from different angles, with the aim of contributing various aspects to the process.

8.1.1. Arabic Sentiment Corpus. To let the work of this dissertation begin, we need to have data that contain sentiment labels in Arabic text. As made clear in Chapter 2, there is a limitation of resources in the Arabic sentiment corpus. Therefore, the first aim of this work is to enrich the field of Arabic sentiment analysis with a new labeled corpus. This corpus is built using different domains including user reviews in different areas and different newswire text. The annotation process is performed on both the document and sentence levels. Each dataset is labeled with its sentimental orientation that involves subjective {positive, negative, or neutral}, or objective.

8.1.2. Feature Extraction. After the dataset is prepared, the next step of sentiment analysis begins. This step firstly includes creating, extracting, and finding the best features that work well with the sentiment classification tasks. These features involve different types such as n-gram models, adding Part-Of-Speech (POS), using ADJ_ADV (Adjective, Adverb), using Based Phrase Chunk BPC, calculating Polarity Score, number of sentences or words, sentence location, and word cluster. Different ML classifiers are investigated during this task including Multinomial Naive Bayes (MNB), Support Vector Machine (SVM) with linear kernel, and Neural Networks (NNs).

In the sentiment analysis field the reasonable accuracy that is recorded in English was around 70% (Pang and Lee, 2008). In our work, the baseline model that includes the only bag-of-word (BOW) model achieves a performance of 35% - 83% at the sentence level and

49%-96% at the document level for Arabic text. Using more n-gram models such as bi-gram model alone has not increased the accuracy of the classification. However, containing different n-gram models together such as using uni-gram with bi-gram model improve the performance compared to using the baseline model alone. For example, the performance increases to 67% from 65% using a combination of n-gram models with the document level classification in the restaurant reviews domain.

Adding the morphology features helps the classifier. This features help to distinguish between different words that have the same letters but play a different roles in the sentence. The POS tag tends to help improve the performance of sentiment classification. In addition, using BPC gives the classifier the ability to capture the sentiment in the phrase level instead of the word level. Using this feature also tends to increase the accuracy of the classifier. Using BPC improves the polarity_3 classification that have three categories of polarities (positive, negative and neutral) by 4% and 8%, Tables 4.16 and 4.16, with movie reviews as well as the other morphology features POS and Adj_Adv, Tables 4.9 and 4.10. In addition, the performance jumps to 92% and 93% using either POS or Adj_Adv respectively in document level binary polarity classification with market reviews, Table 4.10. These results suggests that the morphology features play significant roles in Arabic sentiment classification.

An other type of feature created here is the stylistic features. This type includes different features such as the number of sentences or words in the text and the position of opinioned sentences in the document. The first feature captures whether the short or long text is subjective or objective. We have noticed that the number of words or sentences feature has an impact on the process of classification especially in the subjectivity classification. For example, the performance rises to 72% from 67% at the sentence level subjectivity

classification with news reviews, Table 4.11. This feature plays a central role in the case of all different types and the level of classification except in the case of ternary polarity classification.

The sentence position feature aims to capture the actual sentiment orientation of the text by assuming that the location of this sentiment would be located in the first and the last parts of the text. This feature works well in the case of subjectivity and binary polarity, but it does not add any great impact to the ternary polarity. This may come from the increase in the neutral class within other two categories. This type of class may be found in the middle of the text. We notice that this feature works well for long text to eliminate unnecessary text that might not contribute to the overall sentiment of the text. The result of the classifier was improved in the case of two long type domains that are movie reviews and newswire.

Two additional features were considered that represent semantic information. The score polarity feature is the first one. In this method, the polarity score of the Arabic text would be calculated using specialized SentiWordNet lexicon. This lexicon provides values that determine the positive, negative, and objective orientation of the particular word. After the calculation process is done, new features will be added to the feature model. These features show the average of positive, negative, neutral, or objective score of a particular Arabic text. Due to the limitation of the SentiWordNet in Arabic language, the translation mechanism to English with different stem settings are used to determine these values. The details of this approach are explained in Chapter 4. The results of these two proposed features show the feasibility of using these methods. There is not a significant improvement in the results compared to the baseline model, but we noticed that there was a slight improvement in most cases. In document- and sentence-level classification, the performance was increased

by 1.5% and 1.8% with the newswire domain. The lack of significant increase may be due to the translation methods losing some of actual sentiment aspect of the original language due to error in the translation process. Therefore, we believe that this feature might still have a significant impact on Arabic sentiment analysis.

The word clustering ID is the second semantic feature that is proposed as a new feature. Using word clustering as a feature has shown promise in different NLP task such as Name Entity Recognition (Ratinov and Roth, 2009; Tkachenko and Simanovsky, 2012). This intuition is due to the fact that the word cluster technique will group the words that have the same semantic orientation including the same type of name entity. With the same intuition, this may group all words that have the same sentiment orientation, as explained in Chapter 4. In order to build this feature the Brown Word Clustering algorithm (Brown et al., 1992) is used in our dataset. The word cluster bit-string ID is used as a feature. The results of using this feature show that the impact on the performance of the classifier was minor. Among the best cases, the performance was improved by 3% in the case of the sentence level binary polarity classification with market review as well as in ternary polarity. In other instances, the results were the same as the baseline model or lower by 1%. Our hypothesis was that this feature would improve performance. We believe the drawback of this proposed feature came from the way we applied the word cluster algorithm. We used the word clustering in each data domain separately. In order to a get more accurate word cluster that preserves the semantic orientation of different words, we need a larger amount of data. Finally, the two previous proposed approaches seem promising to add with Arabic sentiment analysis especially with the Dialect Arabic type that has an absence of basic NLP tools.

In general, the results of experiments with various features suggest the following guidelines. Whenever the performance of the different classifiers improved using a specific feature, it gives strong evidence that the particular features are robust and meaningful to Arabic sentiment analysis task. Whenever the features do not improve the different classifiers, it indicates that particular feature is not robust or relevant to Arabic sentiment analysis. Lastly, features might be useful but less robust whenever these features generate mixed performance with the different classifiers.

8.1.3. Machine Learning Classifier. There are also different ML classifiers that are used and investigated in this work. In the beginning, the MNB was compared with the linear SVM. In the baseline experiment, Chapter 4, the SVM achieved better performance compared to the MNB in most cases. Therefore, only the SVM is used in the semantic features experiments. Comparing the SVM with another classifier, either with different kernel functions of the SVM, or with Neural Networks, we found that the performance has been comparable. The SVM also outperforms the NNs with slight differences. However, the NNs classifier can achieve a better result in the case of the short sentence type of Arabic sentiment analysis, especially with the Dialect Arabic. For example, the results of using the NNs are slightly better than the SVM in the case of subjectivity classification with news review, restaurant review, and market reviews. In addition, it improves the accuracy in the case of the document level polarity_3 classification. In general, the difference in performance between the two classifiers is not significant. In terms of the classification timing, the NNs outperforms the SVM. The SVM needs around 30 times more time than the NNs needs to perform the classification process with new data. Therefore, the NNs classifier is preferable in the case of sensitive systems that need quick answers about the sentiment in a text. The

big difference in the timing comes from the difference in the nature and structure of each classifier.

8.1.4. Negation. The critical factor that changes the original sentimental meaning of the word is the negation. In Arabic sentiment classification, this area has not been investigated thoroughly. The first step in our work with this section involves defining the negation concept and lists in Arabic text. It then shows the importance of negation with the opinioned text. These steps are followed by proposing different methods to deal with negation in Arabic sentiment analysis. Before mentioning the methods, we need to revisit some other points in negation. In order to add negation in sentiment analysis, the negated sentence must be determined. Then, the scope of the negation in that sentence mus be specified. After this is done, we can inject the negation with the feature model by adding the artifact tag "_NOT" in front of each word in the negation scope. Depending on this concept different approaches are proposed to find the ones that work best with Arabic sentiment analysis.

Static or fixed methods are proposed here that include adding negation to the uni-gram model, counting negation items in a sentence, various window sizes and using BPC to capture the scope of negation. Adding negation words only increases the performance compared to the baseline model when the negation words are considered as stop words. Another stylistic feature is used that is based on counting the number of negation words found in the sentence. This increases the accuracy in some instances, especially in the subjectivity classification. It is an easy way to add knowledge about negation to sentiment analysis. It may reveal that the subjective sentences have more negation items than the objective sentences.

By using the previous two methods, we add some knowledge that shows a particular sentence has a negation item. The second step is to combine the knowledge of the negation

scope to the classifier. Different mechanisms are proposed here starting with one word after the negation item and ending with assuming that all words after the negation item in the sentence. In addition, the BPC is used to determine the negation scope in a sentence. By comparing all these methods together, we find that using any of them improves the classification process compared to the baseline model. The best method of negation was found to be the one that is based on the BPC. This method can capture the shallow tree of the sentence structure. As a result of that, the effect of negation would be on the same or the following phrase in which the negation appears. The performance, for instance, jumps to 65.3 from 61.8% on average across different domains.

The last proposed method in the negation process is a dynamic approach. In this process, we try to solve and enhance our assumption in the first method. This assumption supposes every sentence that has a negation item is a negated sentence. This assumption is not always true because some negation items could be used to express a different style. We formalize this issue as a tagging problem, such as the POS problem. In this part, we need a negation label in the data that annotates the text with tags that show the negation items and their scope. After training the sequence labeling algorithm on labeled data, we get a model that could be used in the new text to determine the negation in the sentence. The output would be used to tag all words in the negation scope with "_NOT" while building the feature model. Due to time limitations and effort needed to create that negation corpus, we assume that this system already exists. Depending on that, we annotate a sample of our data with the real negation annotation and compared this method with the previous static methods. We notice that there are some abnormal results, especially in the case of subjectivity classification. However, this method increases the performance in the event of

polarity classification. For example, the accuracy reaches 61% in the case of binary polarity classification with movie review. The abnormality in the results may be due to the small amount of data used in these experiment.

8.1.5. Cross-Domain Method. Chapter 7 investigates the method of learning sentimental knowledge from a particular domain and applying it to another domain. This approach will help to improve the process of sentiment classification especially in the language when the resources of the sentiment task are limited, such as what we have in Arabic. The specialized sentiment corpus in Arabic language is very scarce because sentiment analysis is still in the early stages with this language. Therefore, finding a suitable method that might enlarge the knowledge of the classifier across new domains would be fruitful in this field. We investigate and evaluate performing cross-domain training and testing in the area of Arabic sentiment analysis using two approaches, One_to_One and All_to_One. In the first approach, the ML classifier learns a sentiment model from one domain and applies the learned model to another domain. The second technique trains the classifier on all domains except one and applies the model to the excluded domain. The performance of the cross-domain then is compared to the performance of the classifier with in-domain approach.

The results show that the in-domain outperforms the cross-domain methods. This indicates that the ML classifier should be trained on the same domain that we want to achieve high accuracy in, or we have to apply some adaptation technique during the cross-domain method. Even though there is low performance with cross-domain, there are some promising results that may encourage continuing in this direction. The results improve in the case of using cross-domain method with the long type domain. In the movie reviews and newswire domain, the performance of the classifier improves to around 63% and 72%, respectively,

in the sentence level subjectivity classification. This reveals that the cross-domain method helps the classifier to learn exact sentiment orientation from short or medium domains that have a clear sentiment sentence and can apply this model to a long type domains that have more objective sentences.

8.2. Main Contributions

In summary, this work makes the following contributions:.

The study shows the limitations of resources and methodologies in sentiment classification of Arabic text. The main contribution is the comparative study of features as well as ML classifiers in the case of Arabic language, that is a rich morphology and high inflection language. This work involves both types of Arabic which are the Modern Standard Arabic (MSA) and the Dialect Arabic (DA).

The first contribution is the development of a multi-domain sentiment corpus. This corpus has annotations at different levels (document and sentence) and types (subjectivity, polarity_2, polarity_3). This corpus comprises of 6,267 documents and 33,870 sentences. This corpus will be available freely online.

The dissertation presents and evaluates different types of features designed to capture characteristics of Arabic text related to sentiment analysis. These features include basic and advanced ones. The first contribution in this direction involves evaluation of different essential features with various levels and types of classification with different ML classifiers. The second contribution is proposing and evaluating new feature models that involve BPC, polarity score, and word clustering. The polarity score calculation exploited the English

SentiWordNet and a translation technique. Moreover, the study also exploited the unstructured textual data with the intention of developing and evaluating new features that capture global semantic information between words, by performing word-level text clustering.

This work develops a different Arabic sentiment analyzer by learning different machine learning classifiers, MNB, the SVM and the NNs. This analyzer firstly figures out the subjective text and then finds out the polarity of the subjective text. The class of neutral is also added during the polarity classification process.

The study presents a comprehensive work to in cooperate the negation concept with the Arabic sentiment classification. This includes the following contributions. Firstly, a comprehensive list of the negation items in MSA are generated and extended that to some of DA. Different techniques are proposed and evaluated to deal with the negation and how this concept is added to the ML classifier. Some of these techniques are simple, and some of them use the concept of the sentence structure to capture the scope of the negation in the sentence. Presenting a new method to discover the negation in Arabic text and use this method to enhance dealing with the negation in Arabic sentiment analysis in a dynamic fashion was the other main contribution in the negation concept.

Last contribution of this work represents a new direction in applying the cross-domain technique with the limited resources of the sentiment analysis field, such as Arabic language. Different methods are evaluated by applying cross-domain in Arabic sentiment classification.

8.3. Future Work

The work that has been done in this study is neither optimal nor complete. Advances in science will never stop. The field of Arabic sentiment analysis is still in an early stage. Therefore, there are many different areas of improvement for Arabic sentiment analysis.

Improvement could start from the primary step, by adding new Dialect Arabic sentiment corpus and end with using one domain to predict the feeling on another domain. This section will show some of the important work that might be considered in the future to improve the performance of Arabic sentiment analysis.

The first development should be started to improve the NLP tools of the Dialect Arabic. There are different types of the Arabic Dialect. Each of them has unique vocabularies and structure. Therefore, the linguistic field needs special morphology tagger, parse tree, and negation tagger for each type of Arabic Dialect.

The direction of future actions would enrich Arabic sentiment analysis with a more fine-grain sentiment corpus. Most of the works that have been done in Arabic sentiment analysis have only considered two types of polarity. Our work in this study adds another category to the polarity and it shows that the classification process was complicated because of adding more classes. In some situations, we need to distinguish between the strong positive sentiment, the weak and the typical positive feeling as well as the negative opinion. Building fine-grain sentiment may help create more understanding about the sentiment in Arabic by building the separation line between each category. It may then contribute to differentiating the polarity classes, being positive, negative and neutral. This work could be done manually, but it will take time or may be built automatically by applying the same scale rating that is found with the review if they exist. Another type of the fine-grained opinion corpus is the one that considers emotion instead of polarity feeling. This kind of corpus includes the emotion feeling to annotate the opinioned text instead of using positive, or negative tags. The example of these emotions are {happy, sad, rock, scary ... etc}. To the best knowledge of the author, there is not any specialized Arabic corpus that includes

these emotions. Therefore, this is one of the directions that could be followed to enhance and develop the field of Arabic sentiment analysis.

The chosen features that are used to build the feature model play the main role in the classification process. Therefore, the other direction of future work might be done with the features. In this area, there are different routes that could be followed in the development process.

On of these directions is the one that relates to the clustering feature. In our work, we only used the cluster ID tag of each word and compared the performance of this feature with the BOW model. The results show that there is not much improvement in this method. However, there are different actions that might improve this feature. The first one is to add the POS feature with the clustering ID together. This may add some more information to the clustering method and help to distinguish between the types of words within the same cluster. The second direction would involve the process of the clustering from the beginning. We were applying the word clustering algorithm on our data domain by domain. It might be fruitful to apply the word clustering algorithm on all of our data or find a large Arabic corpus and apply the word clustering on that, and then use these outputs in Arabic sentiment analysis.

The last direction of improvement in this feature would be included the way if using the cluster id with the feature model. Table 8.1 displays some of the words and their cluster tag ID. The first column in Table 8.1 shows the total ID that is generated by the algorithm. The second column depicts the way of pruning some of bit string to reach the best cluster combination. We notice that these words have different cluster IDs. However, all of them convey the positive semantic meaning. Instead of using all bit string in the ID tag such as

Table 8.1: Example of pruning bit string of word cluster ID

Words	Original Cluster ID	After Pruning First 9 bits
(**مميز** /*mmyz*/ 'Distinctive')	1010101111010	101010111
(**رائع** /*rAŷς*/ 'wonderful')	101010111100	101010111
(**أبدعوا** /*ObdςwA*/ 'they innovated')	101010111011	101010111

"1010101111010", we could use only the first nine bit string in the ID. This would put all of these words in one cluster. This is the beauty of the Brown Word Cluster Algorithm that provides a way to go up in the hierarchical clustering levels to determine the degree of the cluster that we want. Finding the best bit string numbers would help to improve Arabic sentiment analysis with cluster feature.

The main problem in sentiment analysis, in general, is the subjective problem. This means one text could be positive in some people's point of view whereas other people may see the same text is negative. Therefore, it is hard to make a consensus that a text has a particular sentiment between people. One of possible directions to increase the performance of the classifier is to simulate the behavior of the human being during the solving of some of problem. In order to get accurate and fair results, more than one classifier should be learned and applied to the problem of sentiment analysis. Some of them might learn different knowledge than the other classifier. After the learning process, we could apply all generated models on a new text and use combining methods that get the output from different classifier to increase the performance of the classification. By applying this approach, we can get the benefits of using different classifiers, that may learn in various ways on various features.

The negation concept in Arabic sentiment analysis is discussed thoroughly in Chapter 5. The foundations have been proposed to add negation to the Arabic sentiment classification. However, there is some possible future work that may be followed to enhance the dealing with negation. That work involves determining the effect of the negation in the text. The

parse tree in NLP shows the total structure of the sentence. In addition, it illustrates the relationship between words and phrases in the sentence. The BPC, which is considered as a shallow tree parser, is used in this study and shows some improvements to the classification process. Knowing the whole structure of the sentence instead of the shallow knowledge may help to get the actual effect of the negation in the sentence. After generating the parse tree of the sentence, the location or the node of the negation items would be determined. Any nodes that are children of the negation item node would be considered in the negation scope. The information could be used to inject the negation to the words in the feature model.

There is a need to build negation detection or tagger to the Natural Arabic language processing. Many NLP applications need this tagger to utilize the generated output in their system and improve the performance. In this study, we proposed a dynamic negation method to inject the effect of the negation with Arabic sentiment analysis. We assumed that the existence of the negation tagger by using some of the manual negated data to prove this concept. The results show the promise of this approach, but we need to investigate the performance of the actual and the real negation tagger system in Arabic sentiment analysis. One future work direction in negation is building the negation tagger. This route may include building negated corpora in Arabic and the system that used this data to predict the negation and its scope in the Arabic text. This direction may also be useful in the case of capturing the implicit negation when the negation items are not used.

One other possible direction to take in the future is working across domains. Regarding the limited resource language, such as Arabic in sentiment analysis, the need for working on cross-domain mechanisms is more than other rich languages. In this study, we present the first step in this direction. In our work, we do not use in adaptation methods or protocols

while the cross-domain experiments are performed between different domains. The results were not fruitful in most cases, so the cross-domain may need a more adaptive approach that can capture the sentiment knowledge on one domain and apply it to another one. One possible adaptation mechanism is to use the extra general sentiment corpus beside the training domain. In our case, we could use the SentiWordNet lexicon to calculate the polarity score of the sentence and use this as a feature of sentiment analysis. This method could provide extra knowledge that might help the classifier to learn and make generalizations about the sentiment problem in Arabic text.

The future work mentioned above is not comprehensive, but it gives some ideas of the possible future actions to take. Moreover, the complexity of Arabic as a target language in sentiment analysis makes these tasks more challenging. These challenges should encourage researchers to become involved in the project of developing ideas to solve these problems.

Bibliography

A. Abbasi, H. Chen, and A. Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems*, volume 26:pages 1–34, June 2008. ISSN 1046-8188.

M. Abdul-Mageed and M. Diab. AWATIF: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 19–28, Istanbul, Turkey, may 2012a. ISBN 978-2-9517408-7-7.

M. Abdul-Mageed and M. Diab. Toward building a large-scale arabic sentiment lexicon. In *Proceedings of the 6th International Global Word-Net Conference, Matsue, Japan*, pages 18–23, 2012b.

M. Abdul-Mageed, M. Diab, and M. Korayem. Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2 of *HLT '11*, pages 587–591, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-88-6.

M. Abdul-Mageed, S. Kübler, and M. Diab. Samar: A system for subjectivity and sentiment analysis of arabic social media. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 19–28. Association for Computational Linguistics, 2012.

A. Al-Subaihin, H. Al-Khalifa, and A. Al-Salman. A proposed sentiment analysis tool for modern arabic using human-based computing. In *Proceedings of the 13th International*

Conference on Information Integration and Web-based Applications and Services, iiWAS '11, pages 543–546, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0784-0.

S. Alhazmi, W. Black, and J. McNaught. Arabic SentiWordNet in relation to SentiWordNet 3.0. *International Journal of Computational Linguistics*, volume 4:pages 1–11, 2013.

A. Aue and M. Gamon. Customizing sentiment classifiers to new domains: A case study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, volume 1, pages 207–218. Citeseer, 2005.

M. Bautin, L. Vijayarenu, and S. Skiena. International sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, pages 19–26, 2008.

K. R. Beesley. Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th conference on Computational linguistics*, volume volume 1, pages 89–94. Association for Computational Linguistics, 1996.

F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V. Subrahmanian. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, pages 203–206, 2007.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

D. Bollegala, D. Weir, and J. Carroll. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1 of *HLT '11*, pages 132–141, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, volume 18(4):pages 467–479, Dec. 1992. ISSN 0891-2017.

J. Carletta. Assessing agreement on classification tasks: the kappa statistic. *Comput. Linguist.*, volume 22(2):pages 249–254, June 1996. ISSN 0891-2017.

P. Chaovalit and L. Zhou. Movie review mining: a comparison between supervised and unsupervised classification approaches. In *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 112–121, jan. 2005.

L.-S. Chen and H.-J. Chiu. Developing a neural network based index for sentiment classification. In *International MultiConference of Engineers and Scientists, Hong Kong*, volume 1, pages 744–749, 2009.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

I. G. Councill, R. McDonald, and L. Velikovich. What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 51–59, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

W. Daelemans, V. Hoste, F. De Meulder, and B. Naudts. Combined optimization of feature selection and algorithm parameters in machine learning of language. In N. Lavra, D. Gamberger, H. Blockeel, and L. Todorovski, editors, *Machine Learning: ECML 2003*, volume

2837 of *Lecture Notes in Computer Science*, pages 84–95. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-20121-2.

K. Darwish. Building a shallow arabic morphological analyzer in one day. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, SEMITIC '02, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

E. Daya, D. Roth, and S. Wintner. Learning to identify semitic roots. In A. Soudi, A. d. Bosch, and G. Neumann, editors, *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 143–158. Springer Netherlands, 2007. ISBN 978-1-4020-6045-8.

L. L. Dhande and G. K. Patnaik. Review of sentiment analysis using naive bayes and neural network classifier. *International Journal of Scientific Engineering and Technology Research (IJSETR)*, volume 3(7):pages 1110–1113, 2014.

M. Diab. Second generation tools (AMIRA 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 285–288, Cairo, Egypt, April 2009. The MEDAR Consortium. ISBN 2-9517408-5-9.

M. Duggan and A. Smith. Social media update 2013. *Pew Internet & American Life Project Tracking surveys*, December 2013. http://www.pewinternet.org/2013/12/30/social-media-update-2013/, accessed 19 December, 2014.

A. El-Halees. Arabic opinion mining using combined classification approach. In *Proceeding The International Arab Conference On Information Technology, Azrqa, Jordan*, pages 264–271, 2011.

I. A. El-Khair. Effects of stop words elimination for arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences*, volume 4(3): pages 119–133, 2006.

M. Elhawary and M. Elfeky. Mining arabic business reviews. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1108 –1113, dec. 2010.

A. Farghaly and K. Shaalan. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, volume 8(4): pages 1–22, Dec. 2009. ISSN 1530-0226.

N. Farra, E. Challita, R. A. Assi, and H. Hajj. Sentence-level and document-level sentiment mining for arabic texts. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1114 –1119, dec. 2010.

H. Ghorbel and D. Jacot. Further experiments in sentiment analysis of french movie reviews. In E. Mugellini, P. Szczepaniak, M. Pettenati, and M. Sokhn, editors, *Advances in Intelligent Web Mastering*, volume 86 of *Advances in Intelligent and Soft Computing*, pages 19–28. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18028-6.

J. Glaser, J. Dixit, and D. P. Green. Studying hate crime with the internet: What makes racists advocate racial violence? *Journal of Social Issues*, Volume 58(1):pages 177–193, 2002. ISSN 1540-4560.

X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

N. Habash, A. Soudi, and T. Buckwalter. On arabic transliteration. In A. Soudi, A. d. Bosch, and G. Neumann, editors, *Arabic Computational Morphology*, volume 38 of *Text,*

Speech and Language Technology, pages 15–22. Springer Netherlands, 2007. ISBN 978-1-4020-6045-8.

I. Habernal, T. Ptáček, and J. Steinberger. Supervised sentiment analysis in czech social media. *Inf. Process. Manage.*, volume 50(5):pages 693–707, Sept. 2014. ISSN 0306-4573.

A. E.-D. A. Hamouda and F. E.-Z. El-Taher. Sentiment analyzer for arabic comments system. *International Journal of Advanced Computer Science and Applications(IJACSA)*, volume 4(3):pages 99–103, 2013.

F. Harrag and E. El-Qawasmah. Neural network for arabic text classification. In *Applications of Digital Information and Web Technologies, 2009. ICADIWT '09. Second International Conference on the*, pages 778–783, Aug 2009.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000. ISBN 0130950696.

S. Khoja and R. Garside. Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University*, 1999.

M. Lamar, Y. Maron, M. Johnson, and E. Bienenstock. SVD and clustering for unsupervised pos tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 215–219, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

D. D. Lewis. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 4–15, London, UK, UK, 1998. Springer-Verlag. ISBN 3-540-64417-2.

M. P. Lewis. *Ethnologue: Languages of the world*, volume 9. SIL international Dallas, TX, 2009.

P. Liang. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, May 2005.

Y. Liu, X. Huang, A. An, and X. Yu. ARSA: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 607–614, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7.

A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

M. McGlohon, N. Glance, and Z. Reiter. Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of Fourth International Conference on Weblogs and Social Media (ICWSM)*, pages 114–121, 2010.

A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc., 2009.

M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, volume 6(4):pages 525–533, Apr. 1993. ISSN 0893-6080.

T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In D. Lin and D. Wu, editors, *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 412–418, Barcelona, Spain, July 2004. Association for Computational Linguistics.

J.-C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. In *Conference of the International Society for Knowledge Organization (ISKO)*, pages 49–54, 2004.

B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, volume 2(1-2):pages 1–135, Jan. 2008. ISSN 1554-0669.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, volume 10 of *EMNLP '02*, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, volume 12:pages 2825–2830, 2011.

A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

R. Prabowo and M. Thelwall. Sentiment analysis: A combined approach. *Journal of Informetrics*, volume 3(2):pages 143 – 157, 2009. ISSN 1751-1577.

K. Priddy and P. Keller. *Artificial Neural Networks: An Introduction*. Tutorial Text Series. Society of Photo Optical, 2005. ISBN 9780819459879.

L. R. Purcell, Kristem. Americans feel better informed thanks to the internet. *Pew Internet & American Life Project Tracking surveys*, December 2014. at `http://www.pewinternet.org/2014/12/08/better-informed/\#fnref-12408-1`, accessed 19 December, 2014.

R. Quirk, S. Greenbaum, G. Leech, J. Svartvik, and D. Crystal. *A comprehensive grammar of the English language*, volume 397. Cambridge Univ Press, 1985.

L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9.

M. Rushdi-Saleh, M. Martín-Valdivia, L. Ureña-López, and J. Perea-Ortega. OCA: Opinion corpus for arabic. *Journal of the American Society for Information Science and Technology*, volume 62(10):pages 2045–2054, 2011. ISSN 1532-2890.

K. C. Ryding. *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press, 2005.

M. K. Saad and W. Ashour. OSAC: Open source arabic corpora. In *6th International Conference on Electrical and Computer Systems (EEECS 10), Nov 25-26, 2010, Lefke, Cyprus.*, pages 118–123, 2010.

T. Segaran. *Programming Collective Intelligence*. O'Reilly, first edition, 2007. ISBN 9780596529321.

K. Shaalan. Rule-based approach in arabic natural language processing. *the International Journal on Information and Communication Technologies (IJICT)*, volume 3(3):pages 11–19, June 2010. ISSN 0973-5836.

A. Sharma and S. Dey. An artificial neural network based approach for sentiment analysis of opinionated text. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, RACS '12, pages 37–42, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1492-3.

R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

K. Sparck Jones. Document retrieval systems. chapter A statistical interpretation of term specificity and its application in retrieval, pages 132–142. Taylor Graham Publishing, London, UK, UK, 1988. ISBN 0-947568-21-2.

I. W. Stats. Internet world users by language. *Miniwatts Marketing Group*, 2013. `http://www.internetworldstats.com/stats7.htm`, accessed 31 March, 2015.

C. Sutton and A. McCallum. An introduction to conditional random fields. *Machine Learning*, volume 4(4):pages 267–373, 2011.

K. Taghva, R. Elkhoury, and J. Coombs. Arabic stemming without a root dictionary. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 1, pages 152–157. IEEE, 2005.

Tashaphyne. Arabic light stemmer, 0.2. 2010, 2010. `https://pypi.python.org/pypi/Tashaphyne/`.

M. Tkachenko and A. Simanovsky. Named entity recognition: Exploring features. In J. Jancsary, editor, *Proceedings of KONVENS 2012*, pages 118–127. ÖGAI, September 2012.

D. Trend. Internet use over time. *Pew Internet and American Life Project Tracking surveys*, 2014. `http://www.pewinternet.org/data-trend/internet-use/internet-use-over-time/`, accessed 19 December, 2014.

A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185, 2010.

P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Twitter. What is twitter? *Twitter*, 2012. `https://business.twitter.com/basics/what-is-twitter/`, accessed 10 November, 2012.

M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo. A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 60–68, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Y. Wilks and M. Stevenson. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Nat. Lang. Eng.*, volume 4(2):pages 135–143, June 1998. ISSN 1351-3249.

T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):pages 399–433, 2009.

W. Wright and C. Caspari. *A Grammar of the Arabic Language*, volume 2. Cambridge: At the University Press., 1898. ISBN 9781616405335.

M. Zuckerberg. One billion people on facebook. *Facebook*, 10 2012. `http://newsroom.fb.com/News/457/One-Billion-People-on-Facebook`, accessed 10 November, 2012.