Arabic Blogging Sentiment Analysis

Lama Saleh Alsudias

A thesis submitted for the requirements of the degree of Master of Science

Computer Science

Faculty of Computing and Information Technology

KING ABDULAZIZ UNIVERSITY

JEDDAH – SAUDI ARABIA

Rajab 1435 H- May 2014 G

تحليل الآراء في التدوين العربي

لما صالح علي السديس

بحث لنيل درجة الماجستير في العلوم (علوم الحاسبات)

كلية الحاسبات وتقنية المعلومات

جامعة الملك عبدالعزيز

جدة – المملكة العربية السعودية

رجب ١٤٣٥هـ - مايو ٢٠١٤ م

تحليل الآراء في التدوين العربي

لما صالح علي السديس

بحث لنيل درجة الماجستير في العلوم (علوم الحاسبات)

إشراف:

أ. د/ فتحي البرعي عبدالقصود عيسى

د/ مصطفى السيد صالح الشربيني

كلية الحاسبات وتقنية المعلومات

جامعة الملك عبدالعزيز

جدة – المملكة العربية السعودية

رجب ١٤٣٥هـ - مايو ٢٠١٤ م

بسم الله الرحمن الرحيم

(و فوق كل ذي علم عليم)

Arabic Blogging Sentiment Analysis

By: Lama Saleh Alsudias

A thesis submitted for the requirements of the degree of Master of Science

Computer Science

Supervised By

Prof. Dr. Fathy Elbouraey Eassa

Dr. Mustafa Saleh (Co-Advisor)

Faculty of Computing and Information Technology

KING ABDULAZIZ UNIVERSITY

JEDDAH – SAUDI ARABIA

1435 H - 2014 G

Arabic Blogging Sentiment Analysis

By: Lama Saleh Alsudias

This thesis has been approved and accepted in partial fulfillment of the requirements for the degree of Master of Science (Computer Science)

EXAMINATION COMMITTEE

	Name	Rank	Field	Signature
Internal Examiner				
External Examiner				
Co-Advisor	Dr. Mustafa Saleh	Associate Professor	Information Systems	
Advisor	Prof. Dr. Fathy Elbouraey Eassa	Professor	Computer Science	

Faculty of Computing and Information Technology

KING ABDULAZIZ UNIVERSITY

JEDDAH – SAUDI ARABIA

1435 H- 2014 G

DEDICATION

Dedicated to

my great parents,

my beloved husband, and my dear daughters.

ACKNOWLEDGMENT

I would like to express my gratitude to Allah (God) for providing me the blessings to complete this work. I also would like to ask Him to ensure that this thesis will be beneficial to other researchers.

To my supervisors, ***Prof. Dr. Fathy Elbouraey Eassa*** and ***Dr. Mustafa Saleh:*** I feel highly indebted to you. I am deeply grateful for your suggestion of this topic, support, comments, and guidance.

To my ***wonderful Parents:*** Words fail me to express my appreciation to you, the best father and mother. I wish to give you all thanks and love for your guidance, advice, and endless support.

Thank you to my wonderful husband, ***Adel***, who was supportive and helpful at every stage of this thesis. My dear daughters, ***Hoor and Farah:*** thanks for your understanding for my being busy, especially in the last days of this work.

To all my family, especially my aunt, my brothers, my sister, and my sisters in law, and my friends: I give you all thanks for your belief in me, your constant support, encouragement, and cooperation at all times.

Arabic blogging Sentiment Analysis

Lama Saleh Alsudias

ABSTRACT

Today, microblogging has become the most popular communication tool among users of social networks. Many users share their opinions on different fields, and those users speak different languages, they are of different ages and education levels. Consequently, opinion mining has become an interesting area of research. Arabic is one of the most used languages in the world. In this thesis, we built a machine learning-based sentiment analysis system for mining and analyzing the Arabic tweets in social networks to determine positive and negative sentiments. We also built an application that determines the percentage of positive and negative opinions based on certain hashtags in specific domains. Regarding sentiment mining and analysis, many points would be addressed: building a corpus for Arabic tweets, filtration of the tweets' tokens, and building a fault tolerance-based classifier. The classification would use a fault tolerance technique and different machine-learning algorithms (Support Vector Machine, Naïve Bayes, and Decision Tree). The prototype of the analysis system would be built and evaluated. The study yielded the following results: the average accuracy of the work based on the voter model is **84.8%**. The application ran on the user's selection of the hashtag name and its domain. The areas assessed were educational, social, economic, sports, and political. It showed the percentage of positive and negative tweets in addition to the number of tweets written in this hashtag and the time it takes to process a calculation.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 2: Background

Chapter 3: Related Work

Chapter 4: Proposed Architecture

Chapter 5: Implementation and Testing

Chapter 6: Evaluation and Comparative Study

LIST OF TABLES

Chapter 2: Background

Chapter 3: Related Work

Chapter 5: Implementation and Testing

Chapter 6: Evaluation and Comparative study

LIST OF ABBREVIATIONS

ABSA	Arabic Blogging Sentiment Analysis
API	Application Programming Interface
AWN	ArabicWordNet
C4.5	It is an algorithm used to generate a decision tree
C5.0	It is an algorithm used to generate a decision tree
CART	Classification and Regression Trees
CV	Cross-validation
DF	Document Frequency
DM	Data Mining
DT	Decision Trees
EM	Expectation–maximization algorithm
FT	Fault Tolerance
K-means	Keep It Simple, Stupid
KDD	Knowledge Discovery from Data
KNN	K-Nearest Neighbors
ML	Machine Learning
MSA	Modern Standard Arabic
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair

NB	Naïve Bayes
Neg	Negative
NVP	N-version programming
OAuth	Authentication protocol
Pos	Positive
SVM	Support Vector Machine
TC	Text Classification
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TM	Text Mining
VSM	Vector Space Model

Chapter I
Introduction

Chapter 1

Introduction

1.1 Introduction

Nowadays, most people participate in the Web to express their opinions, which also gives researchers the opportunity to analyze those opinions [1]. The aim of this process is to give researchers general opinions regarding prevalent items or themes in the huge amount of data available on the Internet. These opinions are important to give customers impressions about products and to let producers know about customers' needs.

There are some properties of the Arabic language that make it more difficult to analyze than any other language. The most common problem is the linguistic value of an Arabic word. One word can be understood to have different meanings [2]. In addition, many dialects are in use in Arabic, and the Romanization of Arabic may occur in a sentence [3].

Microblogging platforms are used by different people to express their opinions on different topics, and these can also be a source of people's opinions. The opinions are many, and they are growing every day, which entails an enormous amount of data. Opinions can also be found in different languages and in different social and interest groups. Twitter is the most popular communication tool for users to share their opinions. Each tweet has up to and including 140 characters, though there are no other constraints on the writing of each tweet.

This research builds a machine learning-based sentiment analysis system for mining and analyzing Arabic tweets in social networks to determine positive and negative sentiments. We chose the Arabic language because few papers focus on it, and we focused on microblogging because it is a new domain in which people can express their opinions.

In addition, we built an Arabic Twitter Mining Application that computes the percentage of positive and negative tweets based on certain hashtags in specific domains. The application will search for all tweets regarding a hashtag, and after computation, it will return to the user the percentages of positive and negative tweets, number of tweets found, and the computation time.

1.2 Objectives

This thesis aims to build a sentiment analysis technique based on machine learning. The system will analyze Arabic tweets in the social networks to determine positive and negative sentiments. The analysis system will be achieved with the following sub-objectives:

- Building an Arabic corpus for Arabic tweets.
- Building a mining technique for Arabic tweets in social networks.
- Building a fault tolerance-based classifier that uses machine learning techniques.
- Building the Arabic Twitter Mining Application.

1.3 Outline of this Thesis

Chapter two provides an overview of data mining, text mining, machine-learning algorithms, cross-validation, fault tolerance, Twitter API 1.1, and the Arabic language. Text mining includes a description of text classification. Machine learning algorithms include the Support Vector Machine, Naïve Bayes, and Decision Tree. In the cross validation section, there is a clarification of the k-fold method, and in the Arabic language section, there is a description of Arabic stop words.

Chapter three presents work related to this thesis. It is divided into work related to Arabic opinion mining, work relating to opinion mining in different languages, and comparisons among existing work.

Chapter four illustrates the reasoning behind the thesis, architecture of the proposed system and the application that using the proposed system, and the detailed design. The detailed design includes the mining process and the Arabic Twitter Mining Application.

Chapter five is about implementation and testing. It discusses tools and technologies, Arabic blogging sentiment analysis, the training phase, the testing phase, and the Arabic Twitter Mining Application.

Chapter six evaluates and compares this study. It contains the classifier performance measure, the evaluation method, the results of the evaluation, the discussion, and the impact of this study's performance on the time.

Finally, Chapter seven concludes this thesis and suggests areas for future work.

Chapter II
Background

Chapter 2

Background

2.1 Data Mining (DM)

"Data mining as a discipline is largely transparent to the world" [4]. In recent years, emerging attention has been paid to the huge amount of data available and the need to turn this data into knowledge. Data mining (DM) is the process used to transform a large amount of data into useful information [5]. The goal of DM is prediction, and this can be done by uncovering hidden information, one step in the process of KDD (Knowledge Discovery from Data). Knowledge discovery consists of an iterative sequence of steps including: data cleaning, data integration, data selection, data transformation, **data mining**, pattern evaluation, and knowledge presentation [5] [6].

DM includes a number of stages, shown in Figure 2.1. First, the researcher must determine his or her objectives, including understanding needs and determining topics. Then, data gathering and analysis involves data access, data sampling, and data transformation. After that, modeling and evaluation includes the creation, testing, and evaluation of a model. Finally, deployment involves the application of the model to new data to generate predictions or estimates of the expected outcome [7] [8].

For the person who analyzes the data, it is better to categorize DM into types of tasks depending on multiple objectives [6] [9]:

- **Classification:** mapping data into a predefined class.
- **Regression:** mapping a data item to a real, valued prediction variable.

- **Clustering:** grouping similar data into clusters.
- **Association Rule Discovery:** producing dependency rules that will predict the occurrence of an item based on occurrences of other items.
- **Sequential Pattern Discovery:** finding rules that predict strong sequential dependencies among different events.
- **Summarization:** mapping data into subsets with associated simple descriptions.

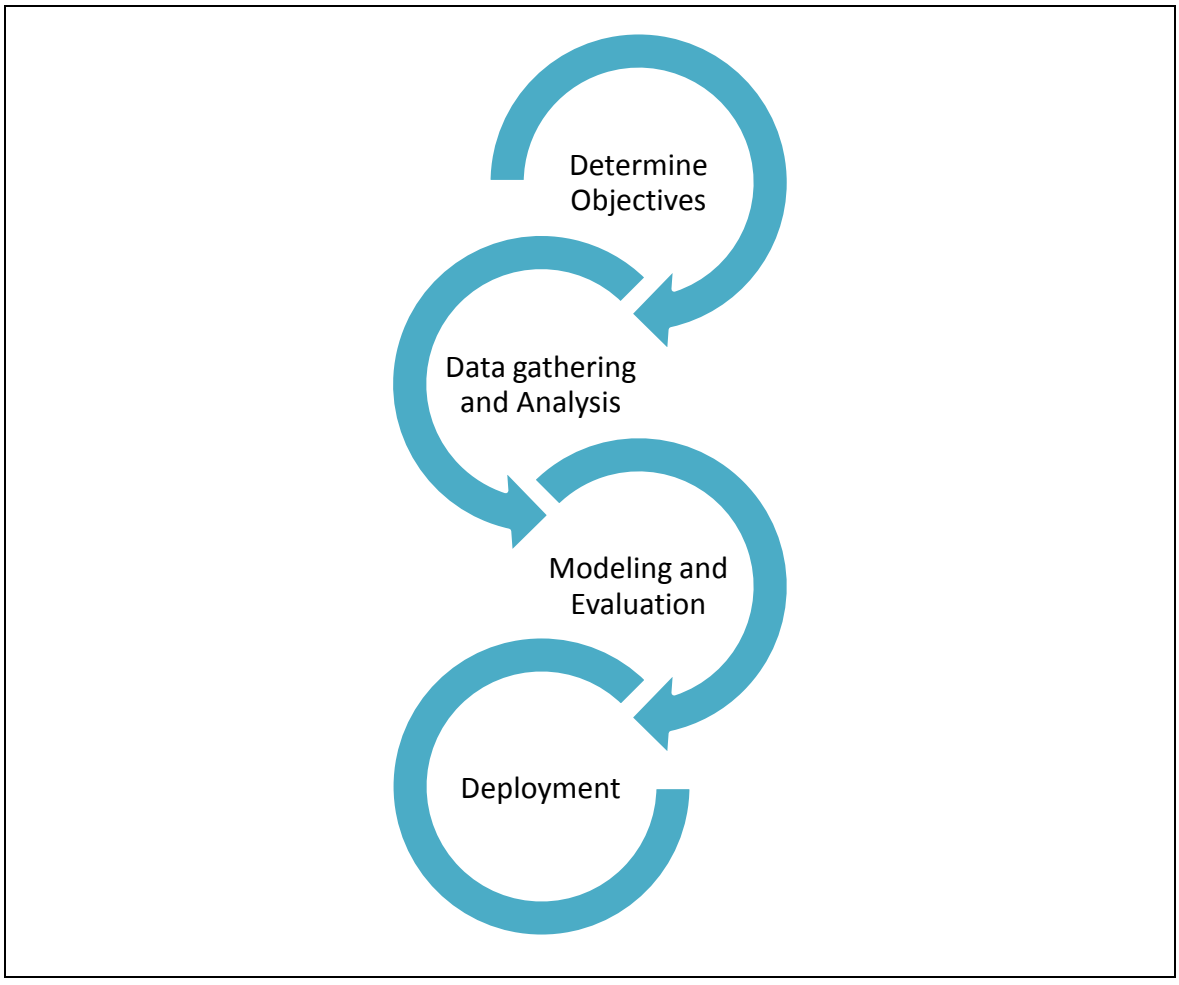


Figure 2.1 The Data Mining Process

2.2 Text Mining (TM)

Text mining (TM) is "the process of extracting interesting information and knowledge from unstructured text" [10]. TM is different from DM because in TM, information is extracted from natural language texts rather than structured databases, which is the case in DM [11].

TM is valuable because much of the world's data can be found in text form (newspaper articles, emails, literature, Web pages, chat conversations, product reviews, etc.). Classification, clustering, and feature extraction have important applications in pure text mining. Other functions, such as regression and anomaly detection, are more suitable for mining mixed data (both structured and unstructured) [8].

2.2.1 Text Classification (TC)

Text classification (TC) is the process of classifying documents into predefined classes. It is also called Text Categorization, Document Classification, and Document Categorization. The goal of TC is to decide the class to which a document belongs [12]. Many applications use TC, like email filtering, Web page classification, newspaper categorizing, Word sense disambiguation, the automated indexing of scientific articles, etc.

There are two approaches to classifying documents: the rule-based approach and the machine learning-based approach. The rule-based approach entails writing a set of rules that classify documents. The machine learning-based approach uses a set of sample

documents that are classified into the classes (training data) and automatically creates classifiers based on the training data [13].

Figure 2.2 shows the steps of the text classification process, which runs in two phases, the training phase and the classification phase. In the training phase, documents are collected in different types. Then preprocessing step is done, including:

- **Tokenization:** converting raw text files into a well-defined sequence of linguistically meaningful units (tokens).
- **Removing stop words:** removing the most frequently used, insignificant words.
- **Stemming:** removing all of a word's prefixes and suffixes to produce the stem or the root.

After that, researchers extract features by selecting the subset of features from the original documents to improve accuracy. At the end of the training phase, documents are classified by assigning labels to documents using machine learning algorithms such as the Bayesian classifier, Decision Tree (DT), K-nearest neighbor (KNN), Support Vector Machine (SVM), Neural Networks, etc. We use SVM, Naive Bayes (NB), and DT, which are described in Section 2.3 [14] [15].

In the classification phase (or prediction phase), new documents will be read, the preprocessing step is done, feature extraction occurs, and finally, classification is performed. The classifier will predict the class of the document depending on the training phase [14] [15].

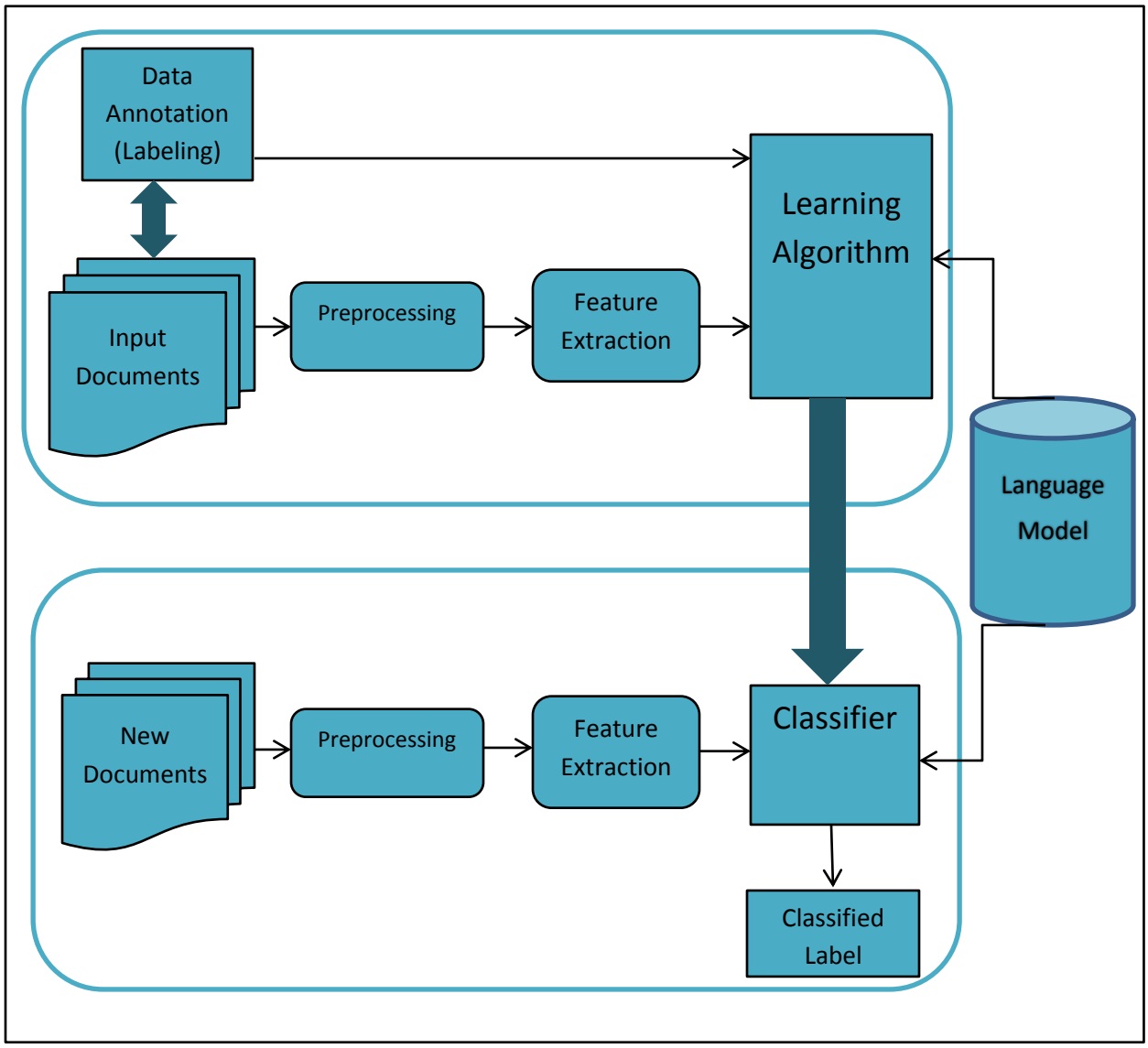


Figure 2.2 The Text Classification Process

2.3 Machine Learning Algorithms

The goal of a machine learning algorithm is to use the training data set to build a model that can classify new documents [15] [16]. There are three types of machine learning algorithms: supervised, unsupervised, and semi-supervised. Supervised machine learning is the task of building a model using labeled training data. Unsupervised machine learning is the task of building a model using unlabeled data. Semi-supervised machine learning is a mixture between supervised and unsupervised classification [14].

There are many examples of machine learning algorithms. The top ten algorithms are C4.5 (C4.5 is an algorithm used to generate a decision tree), K-Means, SVM, Apriori, Expectation–maximization algorithm (EM), PageRank, AdaBoost, KNN, NB, and Classification and Regression Trees(CART) [17]. We focus on the three algorithms that used SVM, NB, and DT, which constitute supervised classification.

2.3.1 Support Vector Machine (SVM)

"SVM is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data" [18]. In other words, given labeled training data (supervised learning), an algorithm outputs an optimal hyperplane that categorizes new examples. A SVM is a decision-based prediction algorithm that can divide data into two classes, based on the concept of decision planes, where the training data is mapped to a higher dimensional space and separated by a plane defining two or more classes of data [19] [20] [21].

Figure 2.3 shows a simple example of a SVM. Given (n) training sets of instance-label pairs (x_i , y_i), while $\{x_1, ..., x_n\}$ are a data set and $y_i \in \{1,-1\}$ are the class label of x_i. The decision boundary should be as far away from the data of both classes as possible. A linear classifier is based on a linear discriminant function of the form [21] [22]:

$$f(x) = w^T x + b \qquad (2.1)$$

Where w is the weight vector and b is the bias (the hyperplane away from the origin). The sign of the discriminant function $f(x)$ denotes the side of the hyperplane. The decision boundary should classify all points correctly:

$$y_i \, (w^T x + b \,) \geq 1 \quad \forall i \quad (2.2)$$

Subject to the decision boundary can be found by solving the following constrained optimization problem:

$$\text{Minimize } \frac{1}{2}||w||^2 \quad (2.3)$$

Minimize (2.3) subject to maximizing the margin:

$$M = \frac{2}{|w|} \qquad (2.4)$$

Maximizing the margin implies that only support vectors are important; other training examples can be ignored. This is the simplest kind of SVM, called a linear SVM [21] [22].

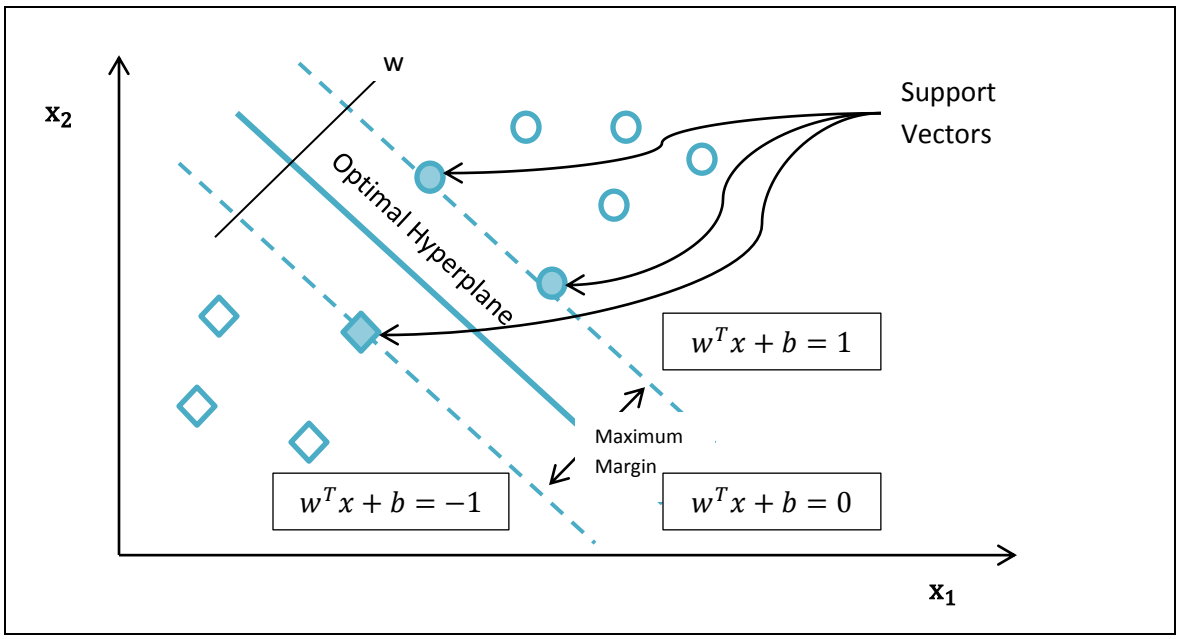


Figure 2.3 A Simple Example of a SVM

A SVM does not need an aggressive feature selection. It can handle even infinitely many features because there are no representation examples in that transformed space. There is only the need to compute the similarity of two examples. Redundant features and high dimensions can be well-handled [23].

Joachims [24] describes why a SVM is good for text classification:

- High-dimensional input space.
- Few irrelevant features.
- Document vectors are sparse.
- Most text categorization problems are linearly separable.

2.3.2 Naïve Bayes (NB)

A NB classifier is "a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions" [25]. Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. The probability that a document (d) belongs to class (c) is calculated as follows [5] [26]:

$$P(c \mid d) = \frac{P(d \mid c)\, P(c)}{P(d)} \qquad (2.5)$$

Where P (d | c) is the probability of generating instance d given class c, P (c) is the probability of the occurrence of class c, and P (d) is the probability of instance d occurring. P (d | c) is difficult to estimate due to the number of possible vectors; d is too high. By using the naïve assumption, the difficulty can be overcome so that any two coordinates of the document are statistically independent [26].

Since P (d) is constant for all classes, we only need to calculate P (d | c) * P (c) [5].

$$P(c \mid d) = P(d \mid c)\, P(c) \qquad (2.6)$$

If we have two classes, c1 and c2, and want to compute if document d belongs to c1 or c2, let us calculate P (c1 |d) and P (c2 |d). When we compare the two results, the higher result means document d belongs to it.

The NB classifier is fast, simple, and computationally efficient; it provides good classification performance. It can be used for both binary and multiclass classification problems. However, it requires a very large number of records to obtain good results.

Naïve Bayes assumes an independence of features, but the solution considers the relationships between attributes.

2.3.3 Decision Tree (DT)

A decision tree (DT) is a graph, where each internal node is a question on features, each branch according to the answers, and each leaf node holds a class label [5]. The steps of classification in a DT are fast and simple. The classifier does not need any domain knowledge or parameter setting. It requires little data preparation, and it is efficient for processing large amounts of training data for DM tasks. It is able to handle both numerical and categorical data (e.g., length < 3 and length ≥ 3) [27].

Figure 2.4 shows a pseudocode of DT algorithm learning. The algorithm builds a tree in a recursive fashion and returns the root. Most DT algorithms are based on a top-down and recursive greedy search for the best decision tree [28].

```
Tree-Learning (TR, Target, Attr)
TR: training examples
Target: target attribute
Attr: set of descriptive attributes
{
    Create a Root node for the tree.
    If TR have the same target attribute value t_i
        Then return the single-node tree, i.e. Root, with target attribute = t_i
    If Attr = empty (i.e. there are no descriptive attributes available),
      Then return the single-node tree, i.e. Root, with most common value of target in TR
   Otherwise
   {
    Select attribute A from Attr that best classifies TR based on an entropy-best measure
    Select A the attribute for Root
    For each legal value of A, v_i , do
    {
       Add a branch below Root, corresponding to A = v_i
       Let TR_vi be the subset of TR that has A = v_i
       If  TR_vi is empty,
           Then add a leaf node below the branch with target value = most
                common value of target in TR
           Else below the branch, add the subtree learned by
                Tree-Learning (TR_vi , Target , Attr -{A})
    }
   }
  Return (Root)
 }
```

Figure 2.4 DT Pseudocode Algorithm

2.4 Cross Validation (CV)

Cross-validation (CV) is a technique that estimates the performance of a model. It splits data into the training data set and testing data and evaluates the risk of the algorithm. The training data set is used for training the algorithm, and the testing data set is used for estimating the risk of the algorithm. The training sample is independent from the testing sample, so CV avoids over fitting [29]. The aim in CV is to ensure that every example from the original dataset has the same chance of appearing in the training and testing set. The common types of CV are K-fold cross-validation, repeated random sub-sampling validation, and Leave-one-out cross-validation [29] [30]. Table 2.1 shows a comparison between different kinds of CV [31].

	Downside	**Upside**
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave -one-out	Expensive Has some weird behavior	Doesn't waste data
10-fold	Wastes 10% of the data 10% more expensive than test-set	Only wastes 10% Ten times more expensive instead of R times.
3-fold	More wasteful than 10-fold. More expensive than test set	Slightly better than a test-set
R-fold	Identical to Leave-one-out	

Table 2.1 Kinds of Cross Validation

The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set, and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. In DM and machine learning 10-fold cross-validation (when k = 10) is the most common. The advantage of k-fold cross validation is that all the examples in the dataset are eventually used for both training and testing [32] [33].

2.5 Fault Tolerance (FT)

The most widely used definition of a fault-tolerant computing system is that "it is a system which has the built-in capability (without external assistance) to preserve the continued correct execution of its programs and input/output (I/O) functions in the presence of a certain set of operational faults" [34]. Fault Tolerance (FT) enables a system to continue its operation rather than failing completely, when some part of the system fails. A problem in a system may occur due to hardware or software failure. Hardware faults are physical faults that can be characterized and predicted. Software faults are logical faults that are difficult to visualize, classify, detect, and correct [35].

There are many techniques involved in software fault tolerance, including traditional techniques like recovery blocks, n-version programming (NVP), n self-checking programming, retry blocks, and n-copy programming. Some new techniques include adaptive n-version systems, fuzzy voting, abstraction, parallel graph reduction, and rejuvenation [36].

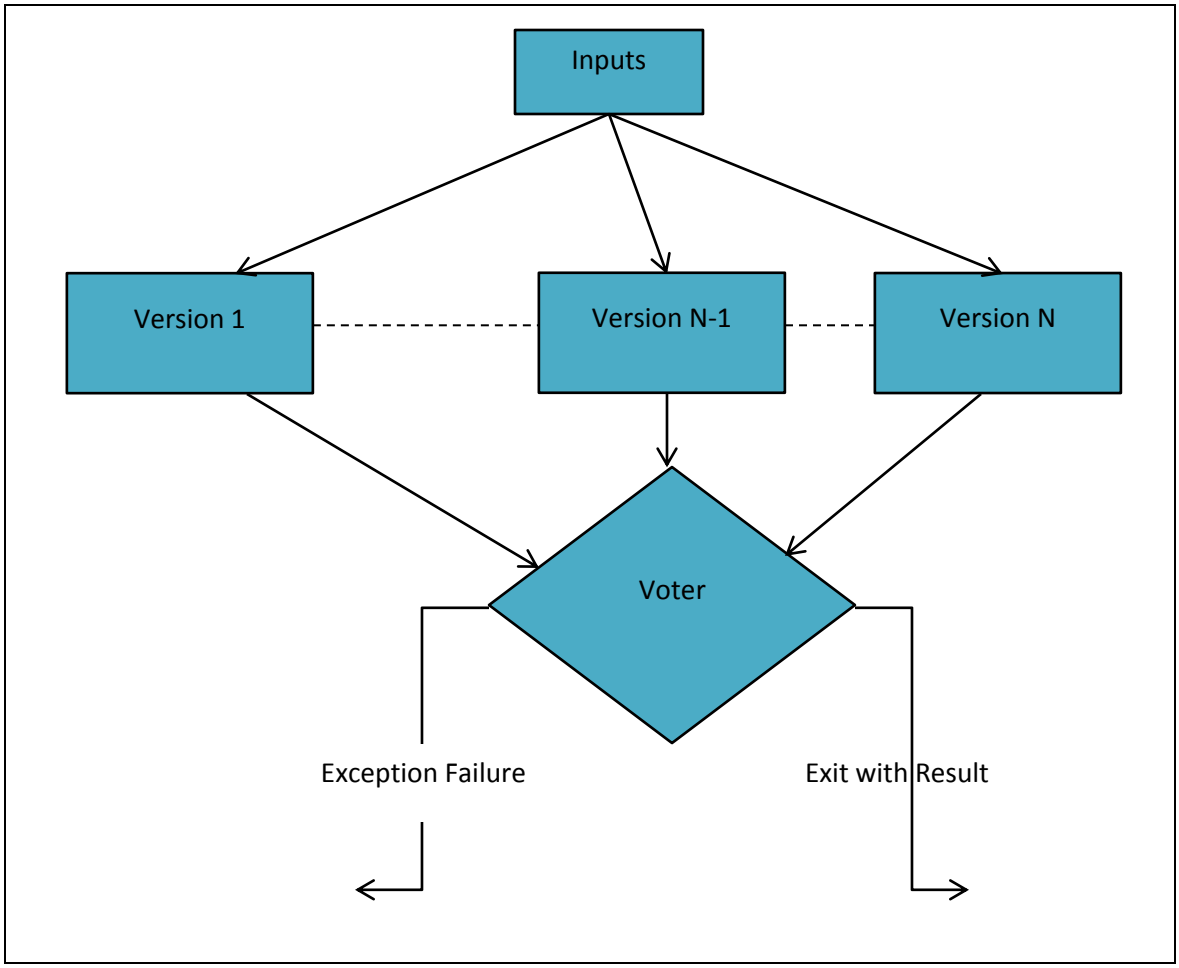


Figure 2.5 N-version programming (NVP) structure.

Figure 2.5 shows the structure of the n-version programming structure that we will use. The task is executed by several programs. The decision or (voter) will determine the best result, or failure exception may occur. NVP (N-Version Programming) enables the parallel execution of different programs, and at least two independents programs must have result [36] [37].

2.6 Twitter API 1.1

Twitter is a microblogging social networking tool that allows users to write short messages (tweets). No more than 140 characters can be in a tweet, including links, Web pages, images, and videos. Following a user in Twitter means can seeing what people write in feed. Unfollowing someone means will stop seeing the tweets of the people that following. Retweeting is sharing a tweet with followers. The hashtag (#) is used to categorize tweets into different topics. When click on a hashtag, all tweets written on that topic will appear [38].

"API" stands for "Application Programming Interface." In Twitter, programmers use API to make applications, websites, widgets, and other projects that interact with Twitter. Users employ http protocol to interact with Web pages. [39] [40]

Twitter API version 1.1 is the update of the Twitter API. Changes in the new version are: JSON support only, authentication is required, improved rate limiting, and changes to the developer rules of the road. In every request to the API, authentication is required on all endpoints. Changes of the rules include that display guidelines will be display requirements, requiring pre-installed client applications to be certified by Twitter, and requiring developers to work with twitter directly if needing a large amount of user tokens [40] [41].

2.7 Arabic Language

The Arabic language is a Semitic language that consists of 28 letters, as follows: أ, ب , ت, ث , ج , ح, خ , د, ذ, ر, ز ,س, ش , ص, ض , ط, ظ, ع, غ , ف , ق , ك , ل , م , ن , هـ , و and ي. There are three vowels, the letters (ا, و , and ي); the rest are consonants. More than 422 million people speak Arabic [42]. Arabic is the language of the Quran (the holy book of the Muslims). Modern Standard Arabic (MSA) is the literary language. It is a pluricentric language derived from the language of the Quran. There are many dialects of Arabic, such as Egyptian, Maghreb, Hassaniiya, Maltese, Sudanese, Levantine, Iraqi, Gulf, Hijazi, Najdi, and Yemeni [42].

Words in Arabic can be nouns, verbs, adverbs, adjectives, or particles. One word may have different meanings, such as (عين) or (Ein), which has more than ten meanings. “Ein” means eye, the essence of a thing, the entirety of a thing, the most important part of a thing, the part of a thing that is currently present, sentinel, spring (water), knee, a non-destructive rain that lasts five or more days, corner, the sun, sun's ray, ready money, gold, a slight imbalance on a scale, Dinar, or seven Dinars. Moreover, one meaning may entail many words. For example, the lexical item “camel” or (جمل) has many words in common use among Arabs [43].

2.7.1 Arabic Stop Words

Stop words are the most frequently used, insignificant words that are useless in information retrieval and text mining. In Arabic, stop words include pronouns, prepositions, adverbs, days of the week, and months of the year. The list of these words

includes (في, من, على, هنا, بين , بعد etc.). It is better to ignore them and index to improve the search.

Chapter III
Related Work

Chapter 3

Related Work

3.1 Introduction

The existing works on opinion mining on microblogging are limited since the phenomenon has only appeared in the last few years. Most research is found at the document level. In addition, few studies focus on analyzing opinions in Arabic.

3.2 Related work on Arabic opinion mining

EL Kourdi et al. in 2004 [44] used the NB algorithm for classifying Web documents into five predefined categories: sports, business, culture and art, science, and health. They used cross validation experiments for evaluation and TF-IDF as the feature selection. The average accuracy found is 68.78% over all categories.

In 2008, Al-Harbi et al. [45] presented a classification on seven different Arabic corpora using SVM and C5.0 (C5.0 is an algorithm used to generate a decision tree) algorithms. Chi-squared statistics were used for feature selection, which is an in-text classification used for ranking features according to usefulness. Two DM software were used, RapidMiner and Clementine. The average accuracy is 68.65% in SVM and 78.42% in C5.0.

Harrag et al. [46] used the decision tree technique to classify Arabic documents. Two different corpora were used, scientific and literary. The researchers used several values for the Term Frequency (TF), Document Frequency (DF), and the combined frequency

(TF/DF). The accuracy was about 93% for the scientific corpus and 91% for the literary corpus. The use of two different corpora allowed them to conclude that a set of factors can affect classifier performances.

A study [1] by El-Haless used the combined classification approach. It utilizes a lexicon-based method to classify documents. The lexicon is built manually based on two resources, the SentiStrength project and an on-line dictionary. The process of filtering is manual. Then the researchers employ the maximum entropy method, which subsequently classified some other documents. Finally, the k-nearest method is used for the rest of the document. Experiments show accuracy moves from 50% when using only the lexicon method to 60%, when using lexicon and maximum entropy, to 80%, when employing the three combined methods.

Saleh et al. made an opinion corpus for Arabic [3]. Then they translated the corpus to English and made a comparison [47]. The data were collected from movie reviews. They used support vector machines and Naïve Bayes for machine learning. They then utilized cross-validation to compare performance. The evaluation was based on three measures: precision, recall, and accuracy. They made a good prepossessing task (tokenizing, filtering, and stemming), and they achieved good results.

Abdul-Mageed and Diab [2] presented a system for subjectivity and sentiment analysis of Arabic social media (SAMAR). The data sets were Dardasha (Arabic for chat), Tagreed (tweeting), Tahrir (editing), and Montada (forum). They adapted a two-stage classification approach. In the first stage, they classified objective from subjective cases, and for the second stage, they classified positive from negative cases. They made

experiments on two standard features, Unique and Polarity Lexicon, and found individualized solutions for each domain and task. They also suggested Arabic features that improved results.

F. Lazhar and T. Yamani [48] identified Arabic opinions in newspapers. This system uses a conceptual model based on the following elements: predicate, source, subject, and model. Then they used XML representation to store opinions. The last step was classification, which did not identify the subject of the opinion.

In 2012, a paper [49] by Ghareb et al. used the Associative Classification approach for mining an Arabic data set. Single-rule prediction and multiple-rule prediction methods were used for classification in this study. The results showed that the model can classify a text data set with a reasonable number of understandable classification rules and produced acceptable accuracy for classifying an Arabic text data set.

Korayem et al. [50] surveyed different techniques for a subjectivity and sentiment analysis of Arabic. They summarized the available resources on Arabic and suggested the method to be followed in building a sentiment analysis system for Arabic. The method includes exploiting wide-scale، domain-specific polarity lexicons and leveraging genre- and social media-specific features. They also suggested a solution that is an alternative to the method of transferring sentiment knowledge from English to Arabic or using language-independent methods.

Shoukry and Rafea [51] work with Twitter data. They use a SVM and NB as machine-learning algorithms and a combination of unigram and bigram as features. The tweets

were analyzed as positive or negative. The accuracy would be 0.65 when using NB and 0.72 when using SVM.

El-Halees in 2012 also worked with Arabic opinion mining [52]. She discussed the problem of Arabic comparative opinion sentences. The experiment was divided into three tasks. First, she identified a comparative from non-comparative statement and got an f-measure of 63.73%, depending on the linguistic classification. Then, she used three machine-learning algorithms (NB, KNN, and SVM) and got, in the best case, a performance of 86.63%. Finally, she used a combined approach of linguistic and machine learning and found an f-measure of about 88.87%.

3.3 Related work on opinion mining in different languages

Pak and Paroubek [53] presented a system for Twitter as a corpus for sentiment analysis and opinion mining in the English language. The system is able to determine positive, negative, and neutral sentiments in a document. The classifier is based on the multinomial Naïve Bayes classifier that uses N-gram and POS-tags as features. They show that the techniques are efficient and suggest using a multilingual corpus of Twitter data to compare results.

Research [54] by Balahur et al. studied opinion mining in newspaper quotations. It presents a comparative study on the methods and resources that can be employed for mining opinions from quotations (reported speech) in newspaper articles. It presents different possible targets and the large variety of affect phenomena that quotes contain. It concludes there is better performance when classifying positive or negative quotes and

that the combined resources produce the best results when a vocabulary-based approach is used.

Vinodhini and Chandrasekaran [55] present a survey study about sentiment analysis and opinion mining. They cover the techniques and methods in sentiment analysis and the challenges that appear in the field.

Miranda-Jiménez et al. [56] revealed a study of the Machine Learning-Based Approach for opinion mining Tweets in Spanish. They examine how classifiers work while doing opinion mining with Spanish Twitter data. They explore how different settings (n-gram size, corpus size, etc.) affect the precision of the machine learning algorithms. The experiments were with Naïve Bayes, Decision Tree, and Support Vector Machines.

A paper [57] by Zhang et al. uses a machine learning technique based on a string kernel for classification of reviews written in Chinese. The study performed experiments with an SVM, NB, DT, and SVM with string kernel method. The study shows how it is effective to use machine learning based on outperforms method.

Kumar and Minz [58] used SentiWordNet to extract sentiment features of words. They used three classification algorithms: NB, KNN, and SVM. This study concluded that scores of sentiment information were successfully integrated for sentiment feature extraction and suggested future work use SentiWordNet to select features for dimensions to be explored for the application of the mood classification of song lyrics, poetry, and other small documents.

3.4 Comparison among existing work

Table 3.1 shows a comparison of work level, the model, the machine, and accuracy between sources [1], [2], [3], [44], [45], [46], [51], and [52]. Regarding the work level, research [1], [44], [45], [46], and [52] focus on the document level while [2] is in a different social medium. Paper [3] includes comments about movies, and [51] is on Twitter. The best results can found in the papers *Opinion Corpus for Arabic* [3] and *Improving Arabic Text Categorization using Decision Trees* [46]. Accuracy is about 90% when using the trigrams model and a SVM [3] and about 93% when using the Decision Trees method [46].

Studies	Work Level	Machine	Features	Accuracy
El-Halees (2010) [1]	documents	Combined (Lexicon + maximum entropy + K-nearest)	TF-IDF	80%
Abdul-Mageed et al (2012) [2]	social media	SVM	Unique + Polarity Lexicon Dialect Gender + User ID + Document ID	50-85%
Rushdi-Saleh et al (2011) [3]	movies comments	SVM - NB	N-grams	70-90%
EL Kourdi et al (2004) [44]	Web documents	NB	TF-IDF	68.78 %
Al- Harbi et al (2008) [45]	documents	SVM and C5.0	Chi-Squared statistics	68.65-78.42%

Harrag et al (2009) [46]	documents	Decision Trees	TF, DF, and TF/DT	91-93%
Shoukry and Rafea (2012) [51]	Twitter	SVM and NB	Unigram + bigram	65.4-72.6 %
El-Halees (2012) [52]	documents	NB, K-nearest neighbors and SVM	POS tags	**F-measure:** 86.63-88.87 %

Table 3.1 A Comparison of the Work Level, Model, Machine, and Accuracy between Papers.

Figure 3.1 shows the comparison of classification accuracy on Twitter on [51] that is in the Arabic language, [53] which is in the English language, and [56] in the Spanish language. It has about 14% better accuracy in the English corpus than the Arabic one. Thus, in our work, we can focus on improving the accuracy of the Arabic language.

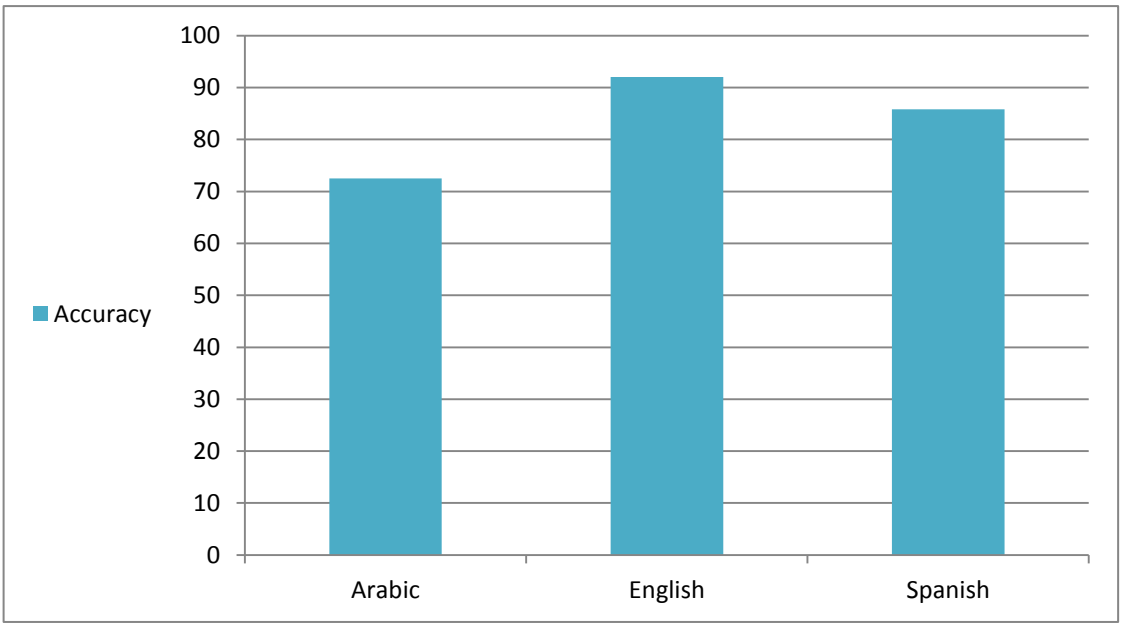


Figure 3.1 A Comparison of the Accuracy between Arabic, English, and Spanish Corpuses on Twitter. (Paper [51] **Arabic, paper** [53] **English, paper** [56] **Spanish)**

Chapter IV
Proposed Architecture

Chapter 4

Proposed Architecture

4.1 Rational Reason

After reviewing most of the work done in sentiment analysis, we found few papers focusing on the Arabic language and the microblogging level.

Today, microblogging has become the most popular communication tool between users of social networks. Many users share opinions on different fields, languages, ages, education level, and the like. Opinions are important for customers and producers. Customers need to have general ideas about products, and producers need to know customers' needs [3]. We chose Twitter due to it being the most popular microblog in which people can write a real-time message about their opinions. There are some challenges, however, regarding a sentimental analysis of Twitter data because of messages' short lengths and irregular structures [59].

We chose Arabic for different reasons. First, the Arabic language is complex and has properties that make it is hard to analyze. Second, there are many dialects in use in Arabic like Saudi, Egyptian, Moroccan, and the like. Third, an Arabic word can have great linguistic variability. One word can be understood to have different meanings [2]. Finally, the Romanization of Arabic may occur in opinions [3].

We needed to choose the best three machine-learning algorithms that give optimal accuracy in DM (data mining). A SVM and NB were identified as among the top-ten

DM algorithms by the IEEE international conference on DM [17]. They are also used in the most recent studies on DM and gave good performance, like [3], [47], [53], [56], and [51]. A SVM has a good performance for text classification on a large data set as in [60], [61], [62], and [63]. NB is fast, space-efficient, not sensitive to irrelevant features, and handles real and discrete data. It shows good accuracies in [64] and [65], which work on Arabic text classification. The learning and classification steps of a Decision Tree Algorithm are simple and fast. Decision Tree can handle high-dimensional data, and it does not require any domain knowledge or parameter setting. It also shows [46] high accuracy in Arabic text categorization.

The three algorithms have some limitations that will affect the performance of classification. A SVM is sensitive to noise. A relatively small number of mislabeled examples can dramatically decrease its performance. One paper [26] concluded that the NB technique gives higher accuracy when combined with other methods. Overfitting may occur in a Decision Tree algorithm when many of branches may reflect anomalies in the training data due to noise or outliers.

We want to enhance accuracy by making the voter model, which is based on the n-version fault tolerance technique on different machine-learning algorithms. N-version programming will accept the voted result after a task is executed by several programs [36]. After obtaining results from three algorithms, they are compared to find the voted result.

Our idea is to design an application with which the user can know the percentage of positive and negative tweets regarding any Arabic hashtag (#). No similar application in

Arabic has been found thus far. Tweets in the education domain are different from the business domain. One paper [66] showed how one word in Arabic has different meanings based on its domain. It gives the example of a word (قاعده) or (قاعدة), which has several meanings: rule, a database, a base, mass of people, and sitting. Accordingly, the user can choose the domain of the hashtag he or she wants to analyze from many domains (social, economic, educational, sports, or political). When the domain is assigned, the percentage of the classified tweets will be more accurate.

4.2 Architecture

To achieve our goal, we made two architectures, one for the mining process and the other for the application that is using the proposed system.

Figure 4.1 shows the basic block diagram of the proposed system. First, we collected Arabic text from Twitter microblogging. Then, we built the corpus by labeling tweets positive or negative. After that, the preprocessing stage included tokenizing, filter stop words, stemming, generating n-grams, and filtering tokens by length. Then classification process was done. It includes three independent machine-learning techniques (Support Vector Machine, Naïve Base, and Decision Tree). Finally, the voter will be producing the correct opinioned text.

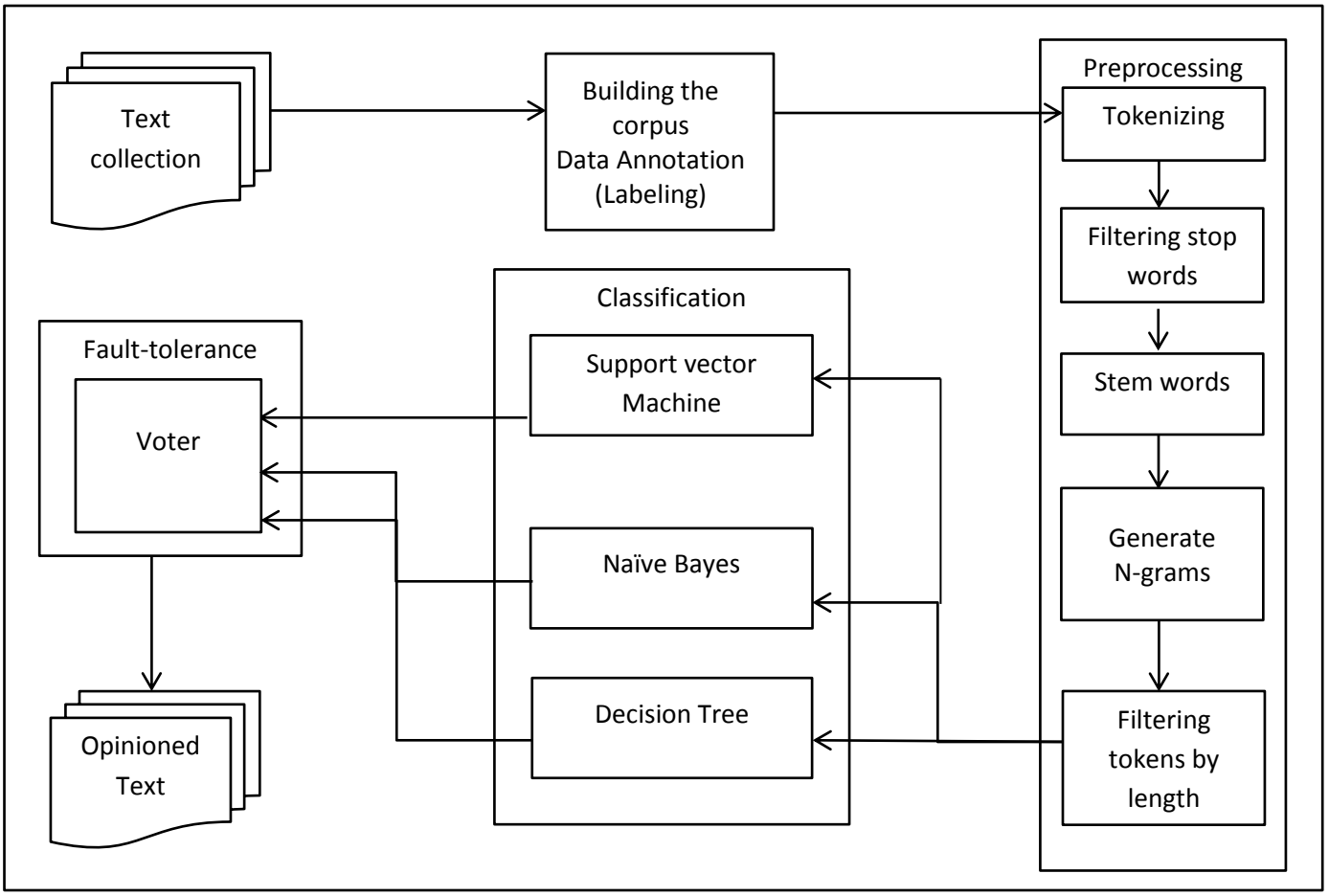


Figure 4.1 The Basic Block Diagram of the Proposed System.

Figure 4.2 shows the basic block diagram of the Arabic Twitter Mining Application. The user enters the name of the hashtag (#) for which the user wants to know the percentages of positive and negative tweets. After determining the domain of the hashtag, the application will calculate the percentages of positive and negative tweets and return them to the user.

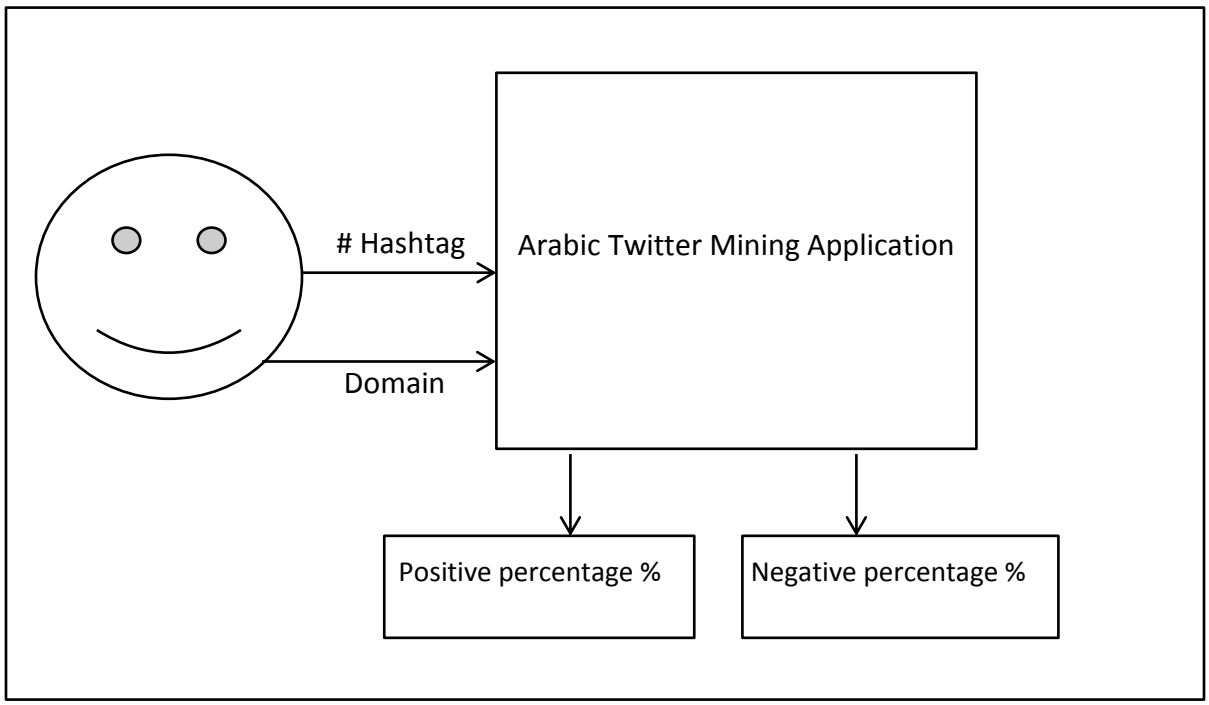


Figure 4.2 The Basic Block Diagram of the Arabic Twitter Mining

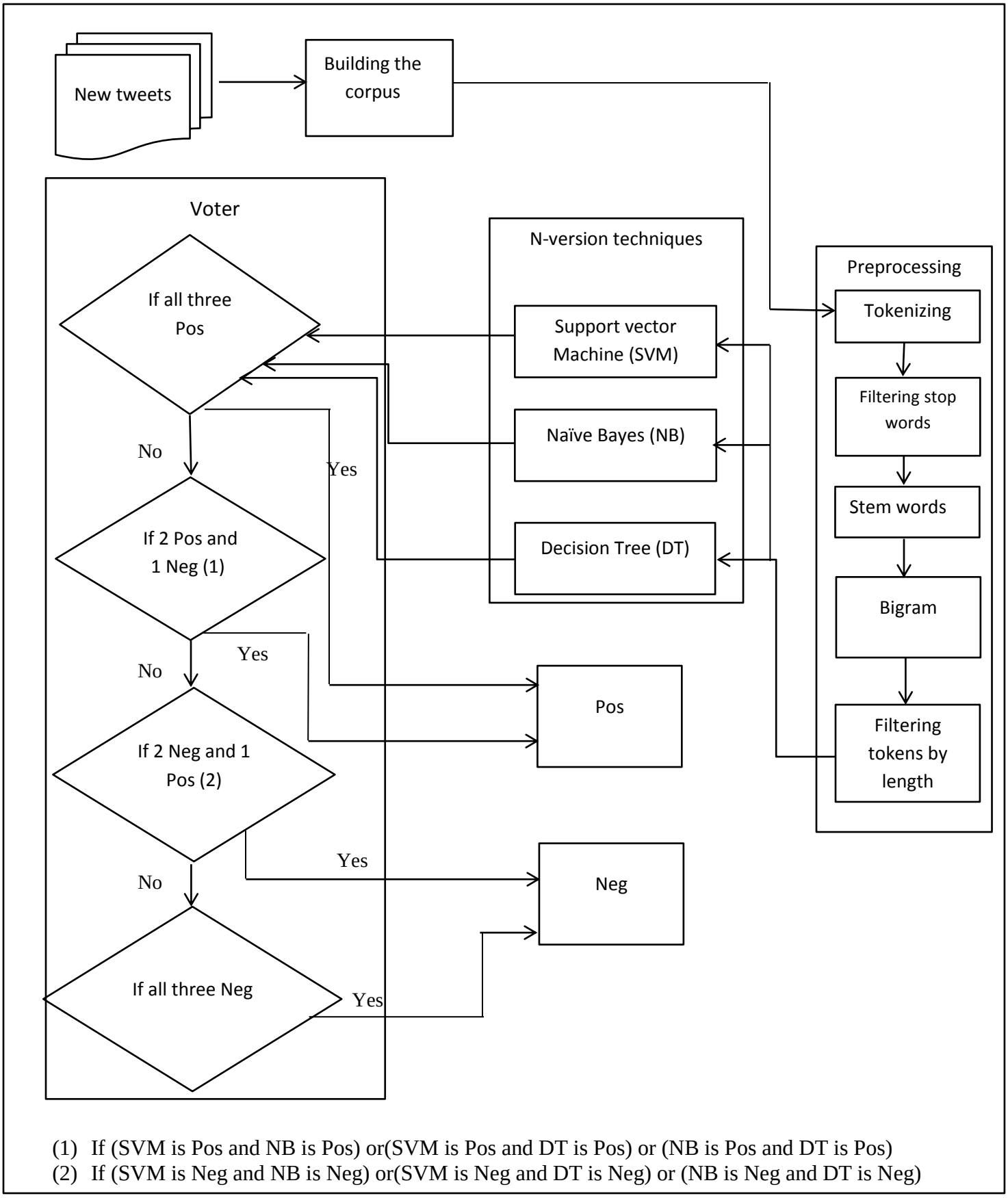
4.3 Detailed design

4.3.1 The proposed system

We wanted to enhance the accuracy of the machine-learning algorithms by making a voter, which works with an n-version fault tolerance method.

Figure 4.3 shows the detailed steps of the proposed system. First is the collection of many tweets from different domains and dialects. The domains include social, economic, educational, sports, and political. The dialects can be Saudi, Egyptian, Kuwaiti, Qatari, Moroccan, Sudanic, and classical Arabic.

Then, tweets were manually labeled positive or negative. We only included tweets with one opinion.



(1) If (SVM is Pos and NB is Pos) or(SVM is Pos and DT is Pos) or (NB is Pos and DT is Pos)
(2) If (SVM is Neg and NB is Neg) or(SVM is Neg and DT is Neg) or (NB is Neg and DT is Neg)

Figure 4.3: The Detailed Diagram of the Proposed System.

After that, preprocessing the tweets includes several steps. Tokenizing splits the text of the document into a sequence of tokens. Filtering stop words removes every token that equals a stop word from the stop words list. Stemming Arabic words uses a stemming algorithm to obtain the root of a word. Generating an n-gram (terms) defines a series of consecutive tokens of double length because in some tweets, the opinion can determine a bigram, for example (أكثر قبحا). Filtering tokens is based on the number of characters they contain.

Then, text is classified by an experiment on different independent techniques: SVM, NB, and DT. The same documents will be classified with different machine-learning techniques. Three different results will be submitted to the voter.

Finally, the voter will produce the final result for each tweet. If all three results are positive, the tweet will be judged positive. If two results are positive and one is negative (If (SVM is Pos and NB is Pos) or (SVM is Pos and DT is Pos) or (NB is Pos and DT is Pos)), the tweet will be considered positive. If two results are negative and one is positive (If (SVM is Neg and NB is Neg) or (SVM is Neg and DT is Neg) or (NB is Neg and DT is Neg)), the tweet will be deemed negative. If all three results are negative, the tweet will also be negative.

4.3.2 Arabic Twitter Mining Application

Let’s assume the user wants to know people’s opinion on any new hashtag that appears. The Arabic Twitter Mining Application can calculate the percentage of positive and negative tweets based on the proposed system.

Figure 4.4 shows a detailed diagram of Arabic Twitter Mining Application. First, the user enters the name of the hashtag about which he or she wants to know the people's opinions. The user also determines the domain of the hashtag: social, economic, educational, sports, or political. Then, the application will collect all the tweets that have the hashtag that user entered. After that, the tweets collected will be entered into the proposed system to classify each tweet's opinion (positive or negative). Finally, the application will calculate the positive and negative percentages and return those to the user.

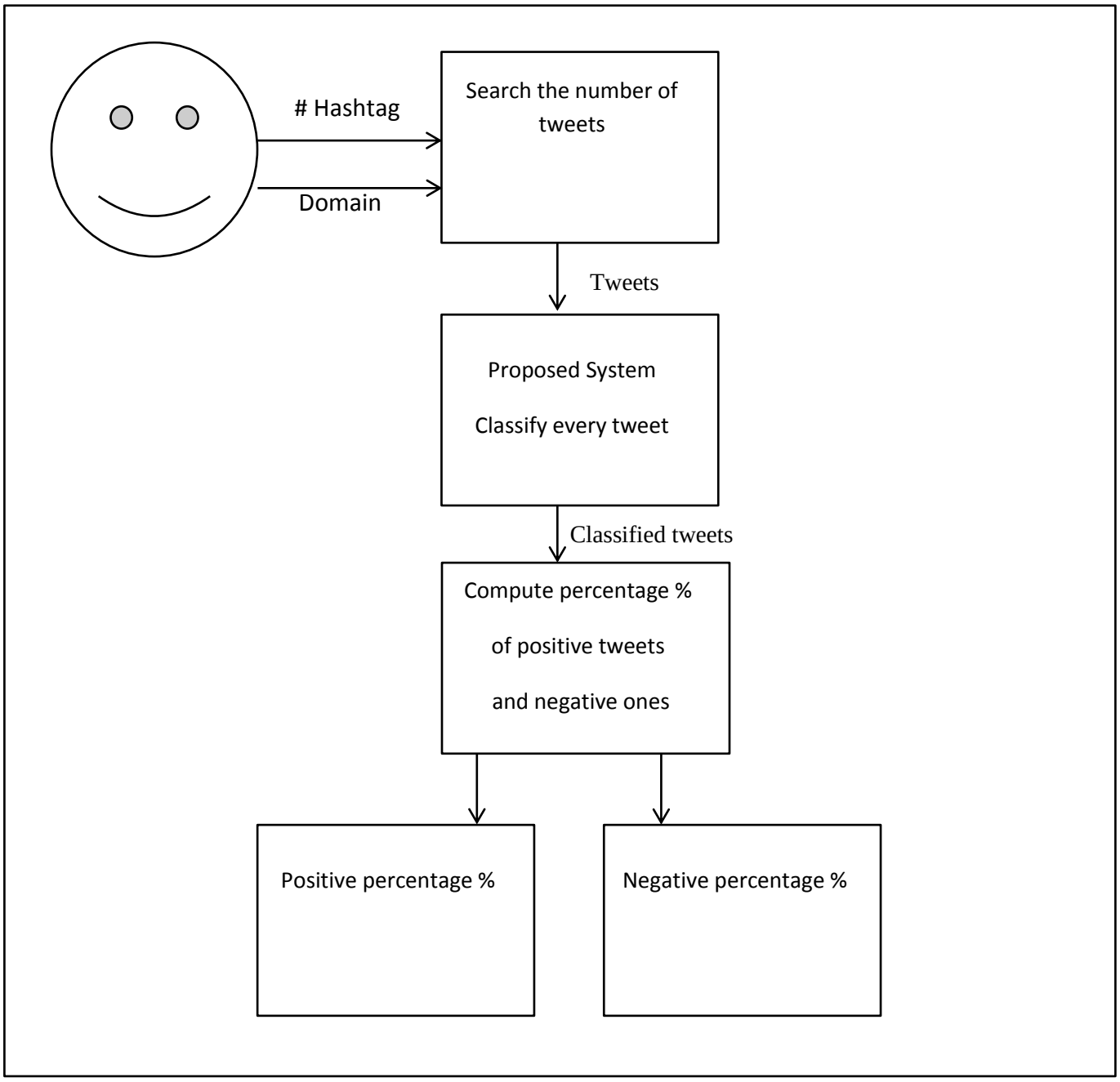


Figure 4.4: The Detailed Diagram of the Arabic Twitter Mining Application

Chapter V

Implementation and Testing

Chapter 5

Implementation and Testing

5.1 Tools and Technologies

5.1.1 Java Programming Language and Eclipse Software

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. We chose it due to its special properties. It is object-oriented, platform-independent, distributed, secure, robust, and multithreaded [67].

Java provides built-in support for multithreaded programming. A multithreaded program contains two or more parts that can run concurrently. Each part of such a program is called a thread, and each thread defines a separate path of execution. A multithreading is a specialized form of multitasking. Multithreading requires less overhead than multitasking processing [67], so we use multithreading to apply the n-version fault tolerance technique.

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages [68].

5.1.2 RapidMiner Software

RapidMiner is a software platform developed by the company of the same name that provides an integrated environment for machine learning, data mining, text mining, predictive analytics, and business analytics [69]. It is used for business and industrial applications as well as for research, education, training, rapid prototyping, and application development. RapidMiner supports all steps of the data mining process, including results visualization, validation, and optimization [70]. RapidMiner is written in the Java programming language and provides a GUI to design and execute analytical workflows. RapidMiner provides data mining and machine-learning procedures including: data loading and transformation, data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment. Finally, RapidMiner supports the Arabic language.

5.1.3 3 Integrating RapidMiner into the Java Application

RapidMiner can be invoked in the Java application [71]**.** We can read the xml files of RapidMiner in Java or construct a process in Java by starting with an empty process and adding operators to the created process [72]. We will create the classification process in RapidMiner and use it in Java; however, we will initialize the voter model in Java.

5.1.4 Twitter4j

Twitter4J is an unofficial Java library for the Twitter API [73]. It can integrate the Java application into the Twitter service. It can work on any Java platform, It is Twitter API 1.1 compatible, and it has built-in OAuth support.

5.2 Arabic Blogging Sentiment Analysis

We use RapidMiner with Java to implement the Arabic Blogging Sentiment Analysis system. It is composed of two main phases, the **Training** and the **Testing** phases. The two phases consist of five steps: collecting tweets, building the corpus, text preprocessing, parallel classification on three techniques, and voter decision.

5.2.1 Collecting Tweets

We make a program that collects tweets using Twitter API 1.1 by twitter4j. The collected tweets are based on certain hashtags. We got more than 7,000 tweets from Twitter from which we have extracted 2,500 tweets consisting of on average 250 positive and 250 negative tweets in each domain. We used 1,500 tweets for the training phase and 1,000 tweets for the testing phase.

5.2.2 Building the Corpus

We made different corpuses based on the domain (social, economic, educational, sports, or political). Table 5.1 shows the hashtags that were used for building the corpus. In each domain, we labeled tweets as positive or negative. Table 5.2 shows some examples of labeled tweets with different hashtags.

Domain	Hashtag
Social	#حملة_الجوازات
	#قيادة_المرأة_للسيارة
	#منع_استخدام_الموظف_جواله_أثناء_الدوام
	#خريجي_الدبلومات_الصحية
	#مشاركة_المبتعثين_بالكريسميس
	#مساجدنا
	#زيادة_الحد_الأعلى_للسرعة
	#مترو_الرياض
	#إثراء_المعرفة
	#الخطوط_ترفع_أسعار_التذاكر_الداخلية
Economic	#تداول
	#الراتب_مايكفي_الحاجة
	#بورصة_الكويت
	#حافز
	#الأسهم_السعودية
	#الوظائف_المستحدثة_لاتكفي
Educational	#جامعة_نورة
	#ثانوية_عامة
	#البحث_العلمي
	#اختبارات_مرحلة_البكالوريوس
	#هل_البيئة_الجامعية_جاذبة
	#cpit49pos[1]
	#مكاتب_خدمات_الطالب
	#يوم_المهنة_الطبي_السادس
	#وفاة_آمنة_بجامعة_الملك_سعود
	#kau_الحاسب_موازي
Sports	#قرعة_دوري_أبطال_أسيا
	#نهائي_كأس_ولي_العهد

[1] This is an Arabic hashtag about a selected topics course at King Abdul-Aziz University.

	#متصدر_لاتكلمني
	#لجنة_تحقيق_لأحداث_مباراة_الشباب_النصر
	#الهلال_الشباب
	#نهائي_كأس_ولي_العهد_الهلال_والنصر
	#صدى_الملاعب
Political	#مع_قطر_ضد_الأخوان
	#رابعة_العدوية
	#في_موزنبيق
	#مرسي
	#وزير_خارجية_قطر_نحن_واسرائيل_أخوة
	#دبي
	#سوريا
	#أحداث_مصر
	#وليا_لولي_العهد

Table 5.1 The Hashtags used for Building the Corpus.

Domain	Label	Tweet
Social	positive	مواطنون: الحملة بدأت تؤتي ثمارها.. والعمل النظامي يضمن حقوق الطرفين #حملة_الجوازات
	negative	#الخطوط_ترفع_أسعار_التذاكر_الداخلية قاتلكم الله ما أخبثكم . . وحسبي الله ونعم الوكيل في كل مسؤول تركنا لجشعكم . .
Economic	positive	اغلقت #بورصة_الكويت على ارتفاع بجميع مؤشراته
	negative	#الأسهم_السعودية تخسر ١,٢% فاقدة ١٠٥ نقاط بنهاية تداولات اليوم
Educational	positive	ما شاء الله تبارك الله نبي نسخه من اساتذتكم شيء جميل ان يكون فيه ماده تستمتع وانت تدرسها #cpit490
	negative	#مكاتب_خدمات_الطالب أكبر معول هدم لتعلمنا العام والعالي
Sports	positive	#الهلال_الشباب تكفون ي الجماهير الهلالية نبي حضور يدعم لاعبينا ويحفزهم
	negative	طفل هلالي يبكي بعد خسارة #الهلال أمام #النصر في #نهائي_كأس_ولي_العهد
Political	positive	#مع_قطر_ضد_الإخوان استقبلت قطر الإخوان والسلفيون وأهل التبليغ وغيرهم من أهل العقيدة السليمة والصحيحة ولم يزدها هذا إلا رِفعة
	negative	الظالم لا يحب أن يشاهد ما يذكره بظلمه !! ومن أجل ذلك فإن أصابع #رابعة تحرق قلوب الظلمة. #أحداث_مصر #رابعة_العدوية

Table 5.2 Some Examples of the Labeled Tweets with Different Hashtags

5.2.3 Text Preprocessing

We completed this step with RapidMiner Software that includes: tokenizing, filtering stop words, finding Arabic stems, generating (bigrams), and filtering tokens by length. Figure 5.1 shows the steps of text preprocessing in the program.

The first step of text preprocessing is the tokenization. Tokenization is the task of converting raw text files into a well-defined sequence of linguistically-meaningful units

(tokens). The tweet is split into a stream of words by removing all punctuation marks, brackets, hyphens, numbers, symbols, and non-Arabic words.

Removing stop words is another common step in text preprocessing. The stop words are the most frequently used and insignificant words, which are useless in information retrieval, and text mining. For Arabic, stop words include pronouns, prepositions, adverbs, days of the week, and months of the year. Stop words are removed because they do not help in determining a document's topic and also for dimensions redaction.

The classification task applied a stemming process in text preprocessing because it makes the tasks less dependent on particular forms of words, as well as reduces the size of the vocabulary, which might otherwise have to contain all possible wards forms.

An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram". We select a bigram model to include tweet like (أكثر قبحا).

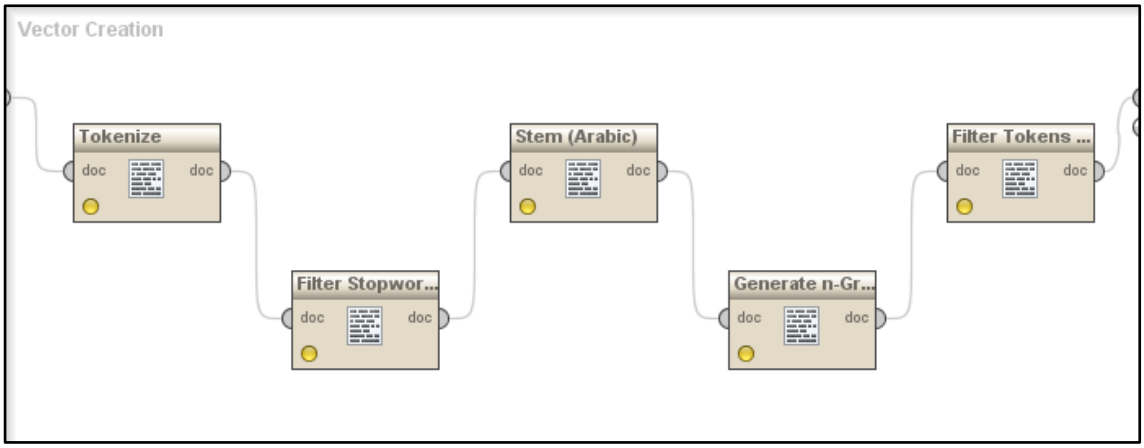


Figure 5.1 Text Preprocessing Steps in the RapidMiner program

5.2.4 Parallel Classification with Three Techniques

We made a Java program that invokes RapidMiner. In Java, we made a multi-thread function, and in RapidMiner, we made classification algorithms. Figure 5.2 shows the prototype of this step. Each thread contained some steps. First, we called RapidMiner and made a classification process with a different algorithm. After classification was done, we computed the time taken for classification because we wanted to compare the parallel process with the sequential one. Finally, we saved the predicted label result to the voter.

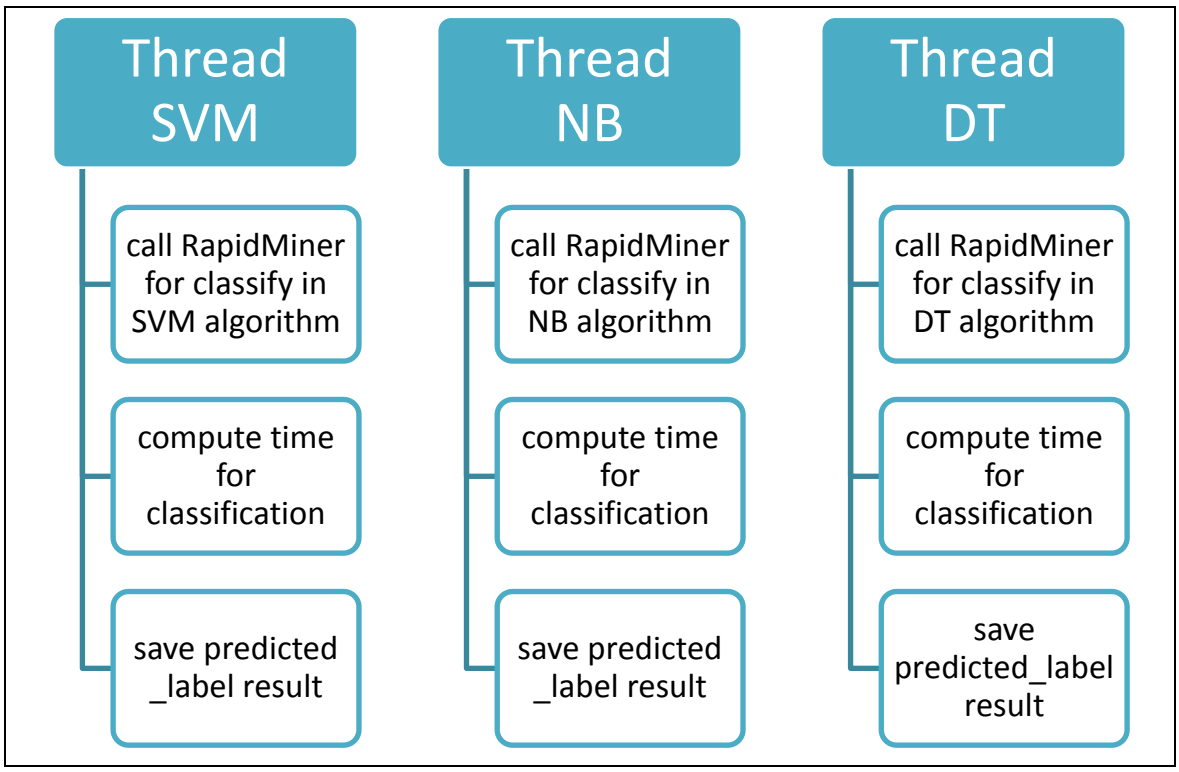


Figure 5.2 Prototype of the Parallel Classification Regarding three Techniques

5.2.5 Voter Decision

After having three results for each tweet from the classification process, the voter will start work. Figure 5.3 shows the prototype for the voter decision. The voter will compute the final predicted label for the tweet. It also computes the total time of the classification process.

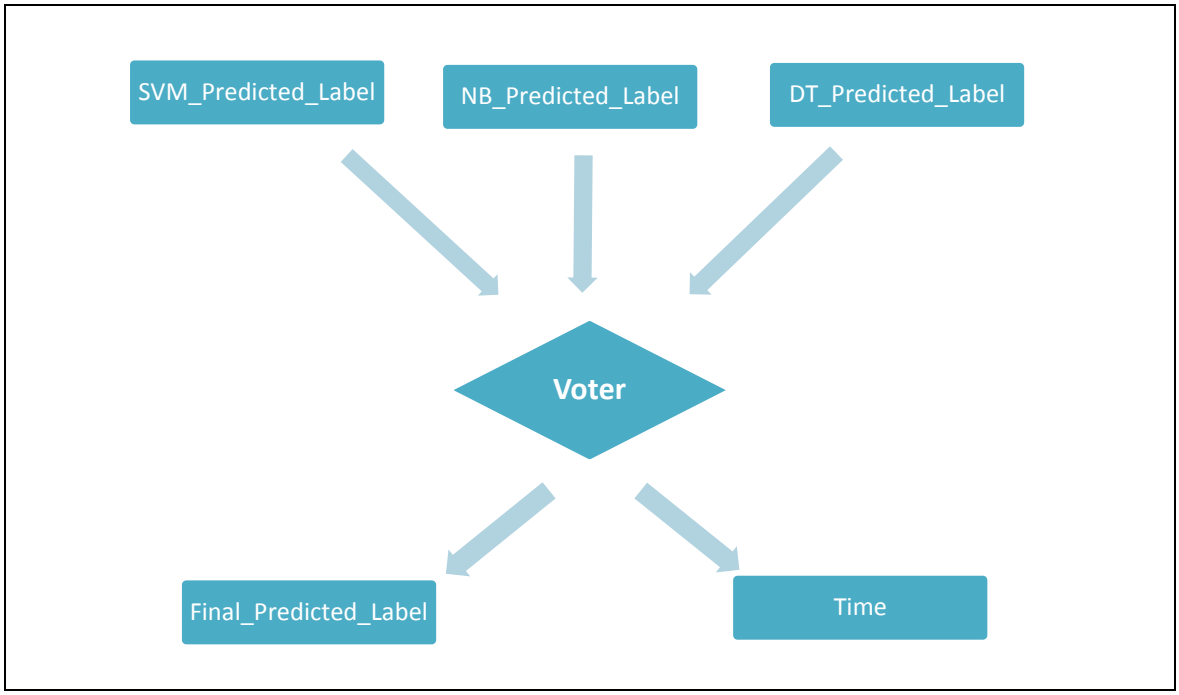


Figure 5.3 Prototype for the Voter

5.3 Training Phase

5.3.1 The voter model

Figure 5.4 shows the pseudo code for how the voter works in the training phase. For each tweet, the voter can determine the final predicted label. We saved the predicted label of all tweets in an array to make the validation process.

5.3.2 X-Validation

X-Validation performs a cross-validation process to estimate the performance of a learning operator. We made tenfold cross-validation. In RapidMiner, the input for the x-validation process is a labeled example set. The x-validation process has two subprocesses: training and testing. The training subprocess returns to the model are usually trained on the input example set. The testing subprocess must return a performance vector. This is generated by applying the model and measuring its performance. The performance vector's attributes include accuracy, precision, and recall.

After classification we have for each tweet:
SVM_Predicted_Label, NB_Predicted_Label, DT_Predicted_Label
Initialize Final_ Predicted _Label
For each tweet:
{
If ((DT_predicted_Label is Pos
and SVM_Predicted_Label is Pos
and NB_Predicted_Label is Pos)
or (DT_predicted_Label is pos
and SVM_Predicted_Label is Pos
and NB_Predicted_Label is Neg)
or(DT_predicted_Label is Pos
and SVM_Predicted_Label is Neg
and NB_Predicted_Label is Pos)
or(DT_predicted_Label is Neg
and SVM_Predicted_Label is Pos
and NB_Predicted_Label is Pos))

Final_Predicted_Label for this tweet is Pos

if((DT_predicted_Label is Neg
and SVM_Predicted_Label is Neg
and NB_Predicted_Label is Neg)
or(DT_predicted_Label is Neg
and SVM_Predicted_Label is Neg
and NB_Predicted_Label is Pos)

```
   or(DT_predicted_Label is Neg
            and SVM_Predicted_Label is Pos
            and NB_Predicted_Label is Neg)
   or(DT_predicted_Label is Pos
            and SVM_Predicted_Label is Neg
            and NB_Predicted_Label is Neg))

                    Final_Predicted_Label for this tweet is Neg
}
Initialize  x-validation process
Set the model to our voter model
Compute performance
Print accuracy, precision, and recall
```

Figure 5.4 Pseudo Code for the Voter in the Training Phase

5.4 Testing Phase

The pseudo code for how the voter works in the testing phase is shown in Figure 5.5. In addition to finding the predicted label of the tested tweet, it will initialize counters to compute the percentages of positive and negative tweets. It will also compute the time of the classification process.

```
After the classification we have for each tweet:
          SVM_Predicted_Label, NB_Predicted_Label, DT_predicted_Label
Initialize Final_ Predicted _Label
Initialize Positive_counter and Negative_counter to zero
For each tweet:
{
If (    (DT_predicted_Label is Pos
            and SVM_Predicted_Label is Pos
            and NB_Predicted_Label is Pos)
    or (DT_predicted_Label is Pos
            and SVM_Predicted_Label is Pos
            and NB_Predicted_Label is Neg)
    or(DT_predicted_Label is Pos
            and SVM_Predicted_Label is Neg
            and NB_Predicted_Label is Pos)
   or(DT_predicted_Label is Neg
            and SVM_Predicted_Label is Pos
```

```
            and NB_Predicted_Label is Pos))

                        Final_Predictred_Label for this tweet is Pos
                        Positive_counter  +=1

if(    (DT_Predicted_Label is Neg
            and SVM_Predicted_Label is Neg
            and NB_Predicted_Label is Neg)
   or(DT_Predicted_Label is Neg
            and SVM_Predicted_Label is Neg
            and NB_Predicted_Label is Pos)
   or(DT_Predicted_Label is Neg
            and SVM_Predicted_Label is Pos
            and NB_Predicted_Label is Neg)
   or(DT_Predicted_Label is Pos
            and SVM_Predicted_Label is Neg
            and NB_Predicted_Label is Neg))

                        Final_Predicted_Label for this tweet is Neg
                        Negative_counter +=1
}

 Percentage_of_positive_tweets   =  Positive_counter / Number_of_tweets * 100
 Percentage_of_negative_tweets  = Negative_counter / Number_of_tweets * 100

 Compute the time of the classification and print it
```

Figure 5.5 Pseudo Code for the Voter in the Testing Phase

5.5 Arabic Twitter Mining Application

The Arabic Twitter Mining Application is an application that can compute the percentages of positive and negatives tweets written with a certain hashtag. The application is written in Java and makes classifications based on the voter model. The user enters the hashtag name and determines the domain of the hashtag. The application returns percentages of positive and negative tweets, numbers of tweets, and the time of the classification. Figure 5.6 depicts the application in detail.

The user can determine the domain of the hashtag that is entered. By default, the domain set includes all the domains of the tweet in the learning phase: social, economic, educational, sports, or political. The program can find all tweets written with a hashtag by using twitter API. Figure 5.7 shows an example of the hashtag (#عودة_الخادمات_الأندنوسيات) discussing the opinions of people regarding a decision. The application found 2,500 tweets with 73.4% supporting the decision and 26.6% against it.

Another example is shown in Figure 5.8 regarding the hashtag (#اللغة_الإنجليزية_بالجامعات). It is in the education domain and discusses taking the English language during the Preparatory Year in universities. The application found 1,078 tweets in which 86.4% were positive and 13.16% were negative.

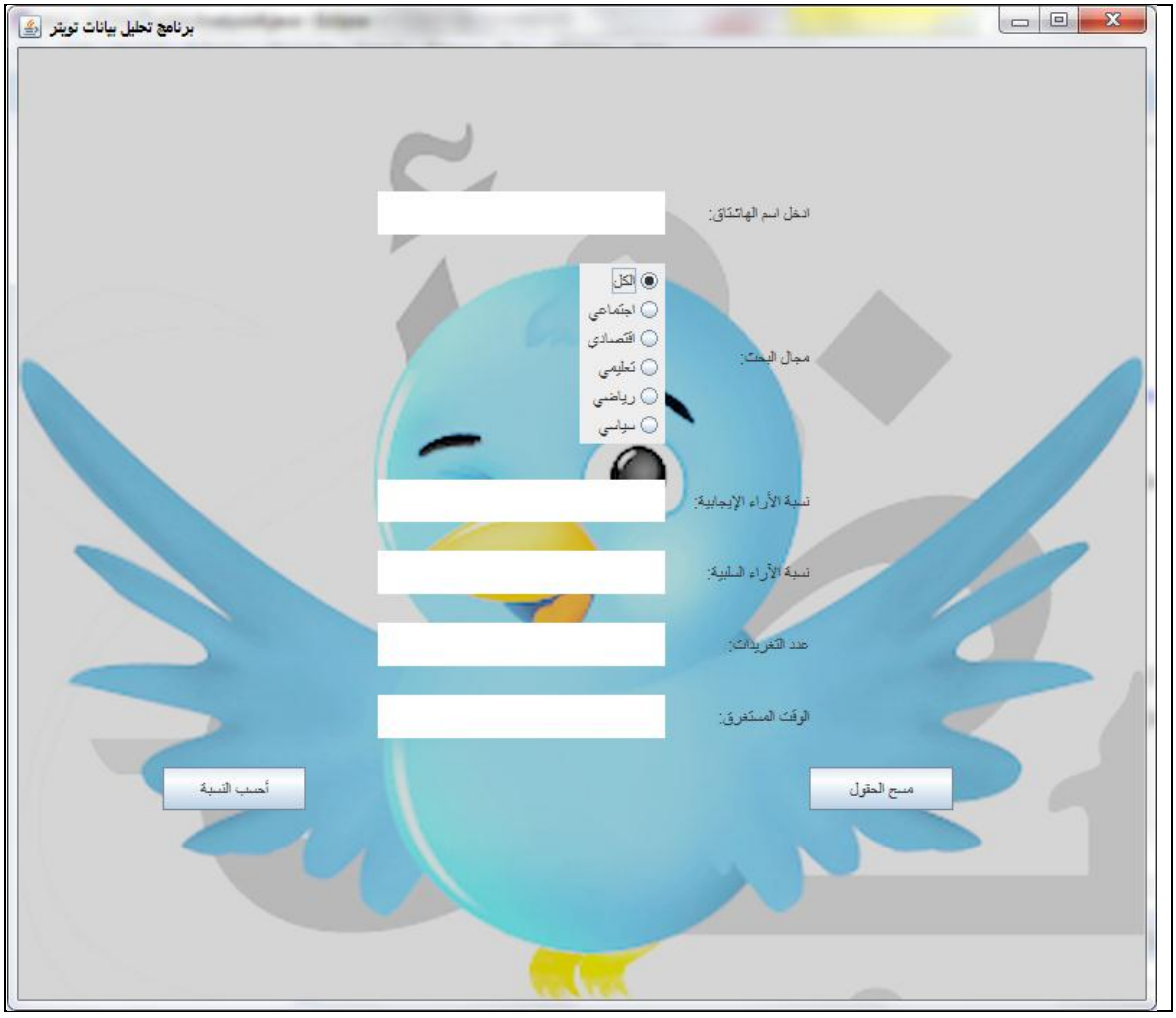


Figure 5.6 Arabic Twitter Mining Application

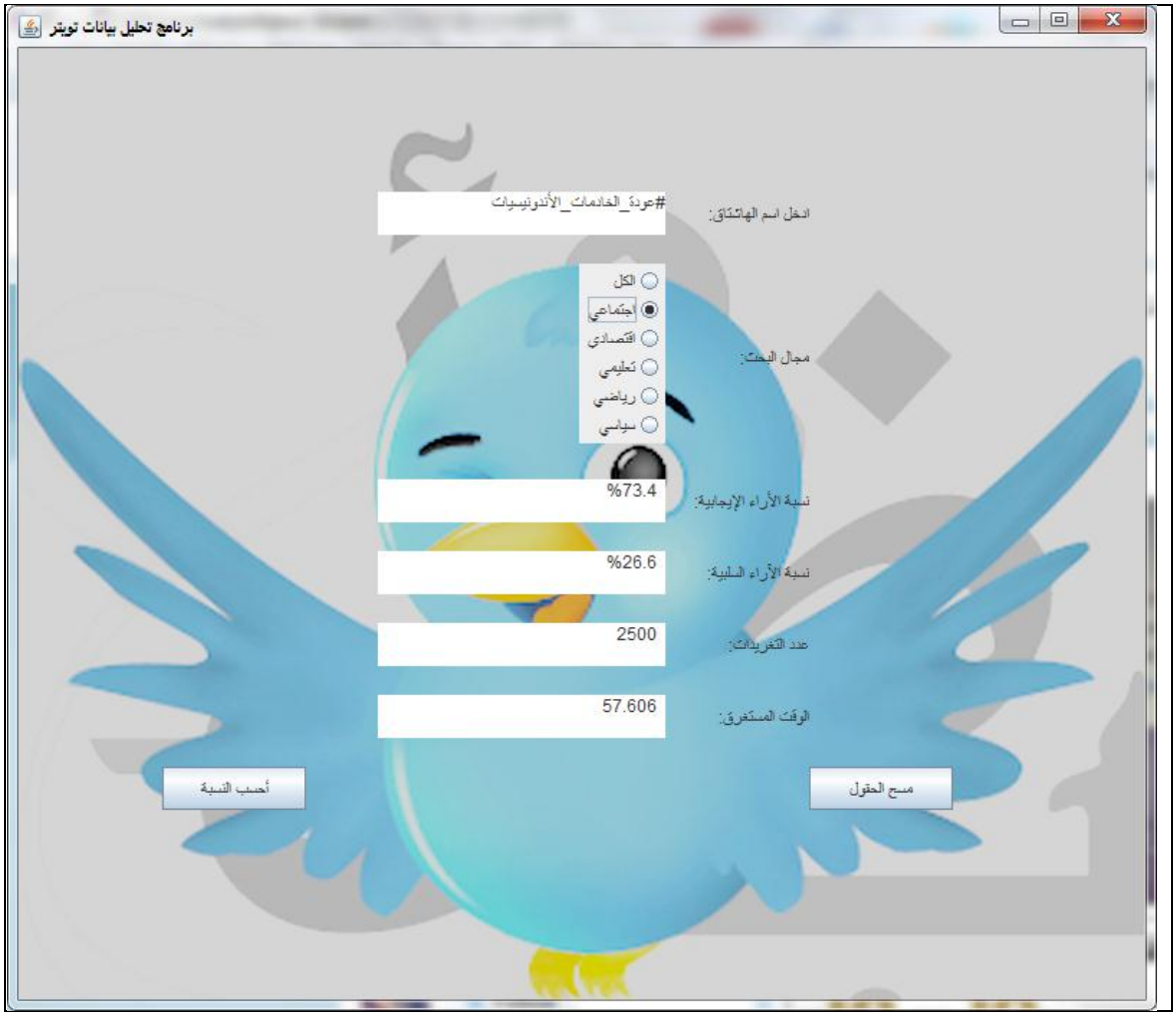


Figure 5.7 Hashtag Example (#عودة_الخادمات_الأندنوسيات)

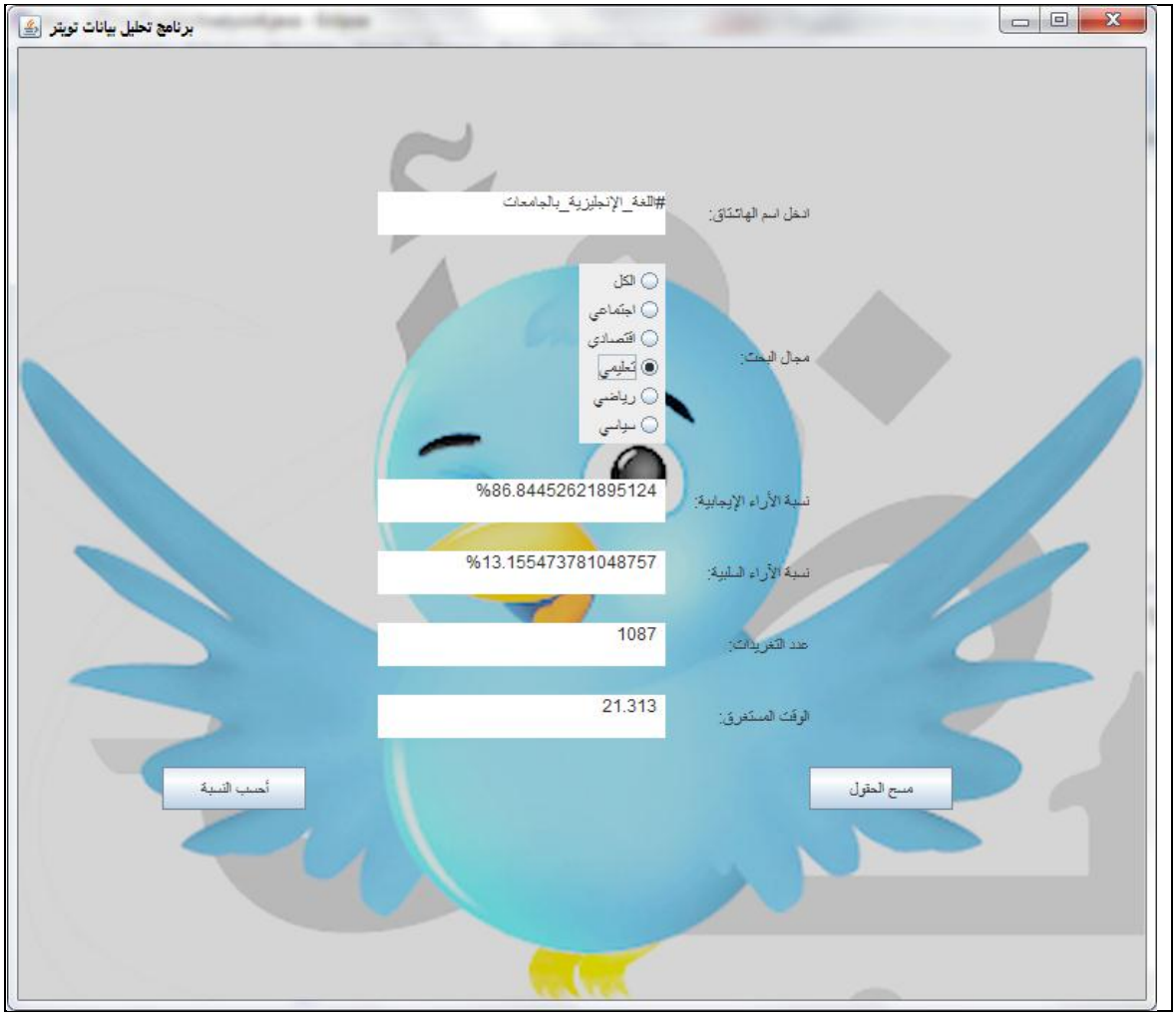


Figure 5.8 Hashtag Example (#اللغة_الإنجليزية_بالجامعات)

Figure 5.9 shows an example of the hashtag (#الهلال) that is in the sports domain. There are 10,000 tweets in it. Of them, 54.8% are positive tweets, and 45.2% are negative ones.

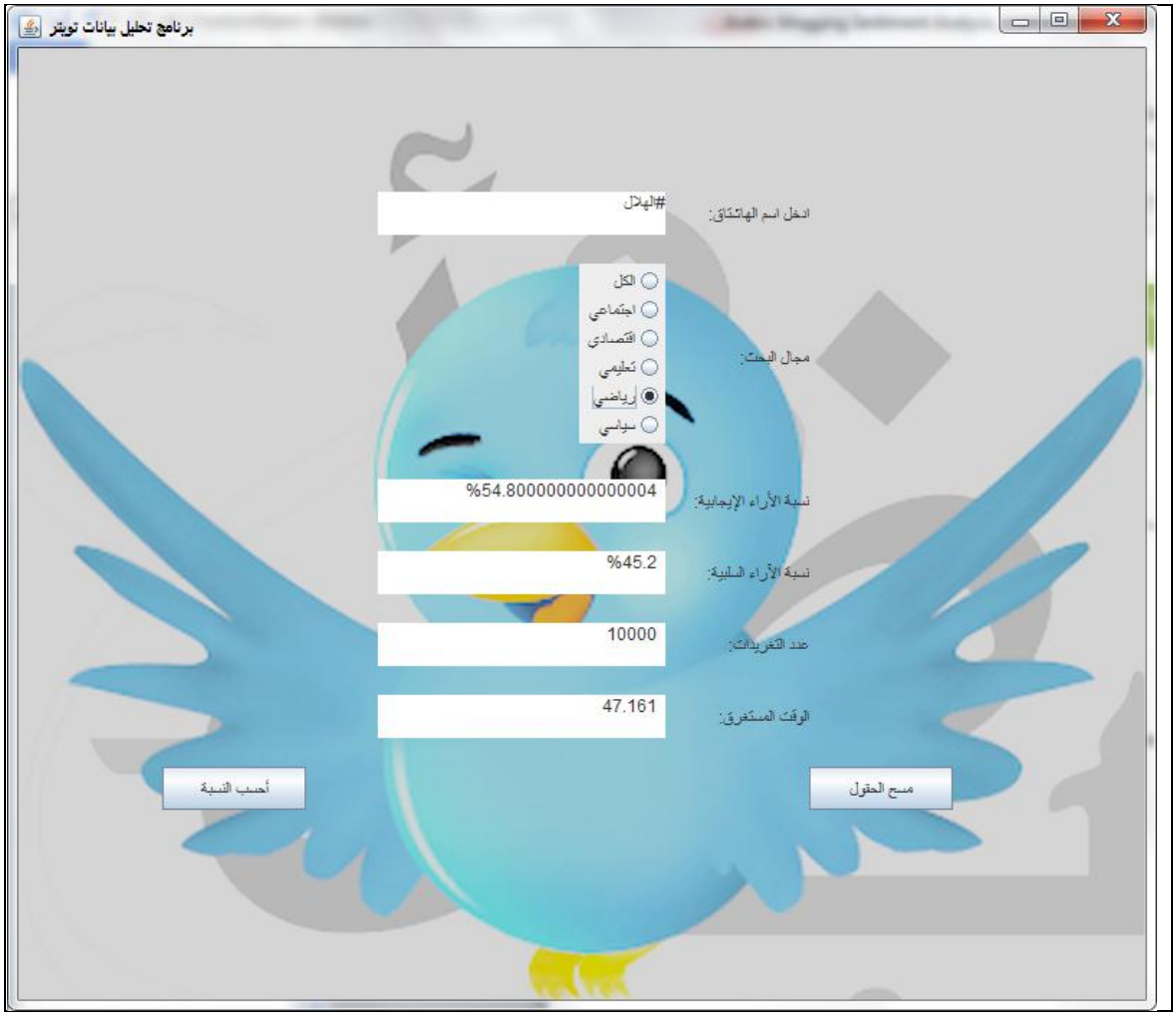


Figure 5.9 Hashtag Example (#الهلال)

Chapter VI
Evaluation and Comparative Study

Chapter 6

Evaluation and Comparative Study

6.1 Classifier Performance Measures

In machine learning, there is a large variation in the measures that are used to evaluate prediction systems. For classification, a variety of measures were proposed, including accuracy, precision, recall, and F-measure [74]. These measures indicate how precise and complete the classification is for the positive class. Table 6.1 shows a confusion matrix that introduces these measures [75].

	Predicted Class	
Actual Class	**Pos**	**Neg**
Pos	TP[2]	FN[3]
Neg	FP[4]	TN[5]

Table 6.1 Confusion Matrix for Two Classes Pos and Neg

[2] True Positive: The number of correct classification of positive samples.
[3] False Negative: The number of incorrect classification of positive samples
[4] False Positive: The number of incorrect classification of negative samples
[5] True Negative: The number of correct classification of negative samples.

The entries in a confusion matrix are integers. The total of the four entries is equal to the number of test samples (TP + TN + FP + FN = N). Depending on the application, many different performance measures can be computed from these entries, such as:

- **Accuracy A** $= \frac{TP+TN}{N}$ (6.1) , which is a percentage of correctly classified data in the tested data set.
- **Recall** $\boldsymbol{R} = \frac{TP}{TP+FN}$ (6.2) , which is the number of correctly classified positive samples divided by the number of positive samples in the data set.
- **Precision** $\boldsymbol{P} = \frac{TP}{TP+FP}$ (6.3) , which is the number of correctly classified positive samples divided by the number of samples labeled positive by the system.
- **F-measure F1** = 2. $\frac{\text{Precicion.Recall}}{\text{Precision+Recall}}$ (6.4) , which is a measure of a test's accuracy. It considers both the precision P and the recall R of the test.

6.2 Evaluation Method

We use the 10-fold cross-validation method for the evaluation. Figure 6.1 shows a visual example of the 10-fold cross-validation method. The data set is divided into 10 portions or (folds). One fold is designated as the test set, while the remaining nine folds are all combined and used for training. Ten iterations will be run for the cross validation. In each iteration, we will compute the confusion matrix entries TP, TN, FP, and FN. The final results will be as follows:

- TP $= \sum_{i=0}^{10} TP_i$
- $TN = \sum_{i=0}^{10} TN_i$
- $FP = \sum_{i=0}^{10} FP_i$
- $TN = \sum_{i=0}^{10} TN_i$

6.3 Results of the Evaluation

Tables 6.2, 6.3, 6.4, 6.5, and 6.6 show the accuracy, recall, precision, and F-measure of the social, economic, educational, sports, and political domains. We can see that the accuracy in the voter model is the best one. The best results regarding accuracy are **94.21** and **92.22,** in the sports and economic domains. Enhancing the accuracy of the voter model in the social domain is clearer than other domains. In the voter model, the accuracy is **71.43,** while it is 67.33, 54.09, and 52 in the other three models. Recall is up to **96.81** of the voter model in the education domain. The best precision for the voter model is **98.41** in the sports domain.

The SVM model has better accuracy than NB and DT, with **93.57** in the sports domain and **90.67** in the economic domain. In the education domain, the accuracy in a SVM is **75,** while it is 60.48 and 56.80 in other models. The recall and precision in a SVM provides good results.

The precision result in NB is very good. It is **84.91** in the education domain, which provides the best result among the models (73.39, voter; 70.19, SVM; and 56.80, DT). Also it is up to **95.04** in the economic domain, which will increase the result of the voter model. NB has accuracy results near the SVM model. For example, in the sports domain, the NB accuracy is **92.82,** and the SVM accuracy is **93.57**.

The recall in the DT model is up to 100 in the social and education domains. The DT model has the worst accuracy in the social, economic, education, sports, and political domains. It is **52** in the social domain and **48.72** in the political domain.

The F-measure gives a better result in the voter model than the other models in the five domains. On average, it is **87.15** in the voter model while it is 85.01, 76.13, and 59.65 in the other models.

Up to 1,500 tweets were found in the domain (all), from the five domains together. Table 6.7 shows the result of the accuracy, recall, precision, and F-measure of this domain. The best accuracy is in the SVM with **81.57,** while in the voter model, it is **80.08**. The best recall is **97.87** in the voter model. The NB model has the best precision with **89.73**.

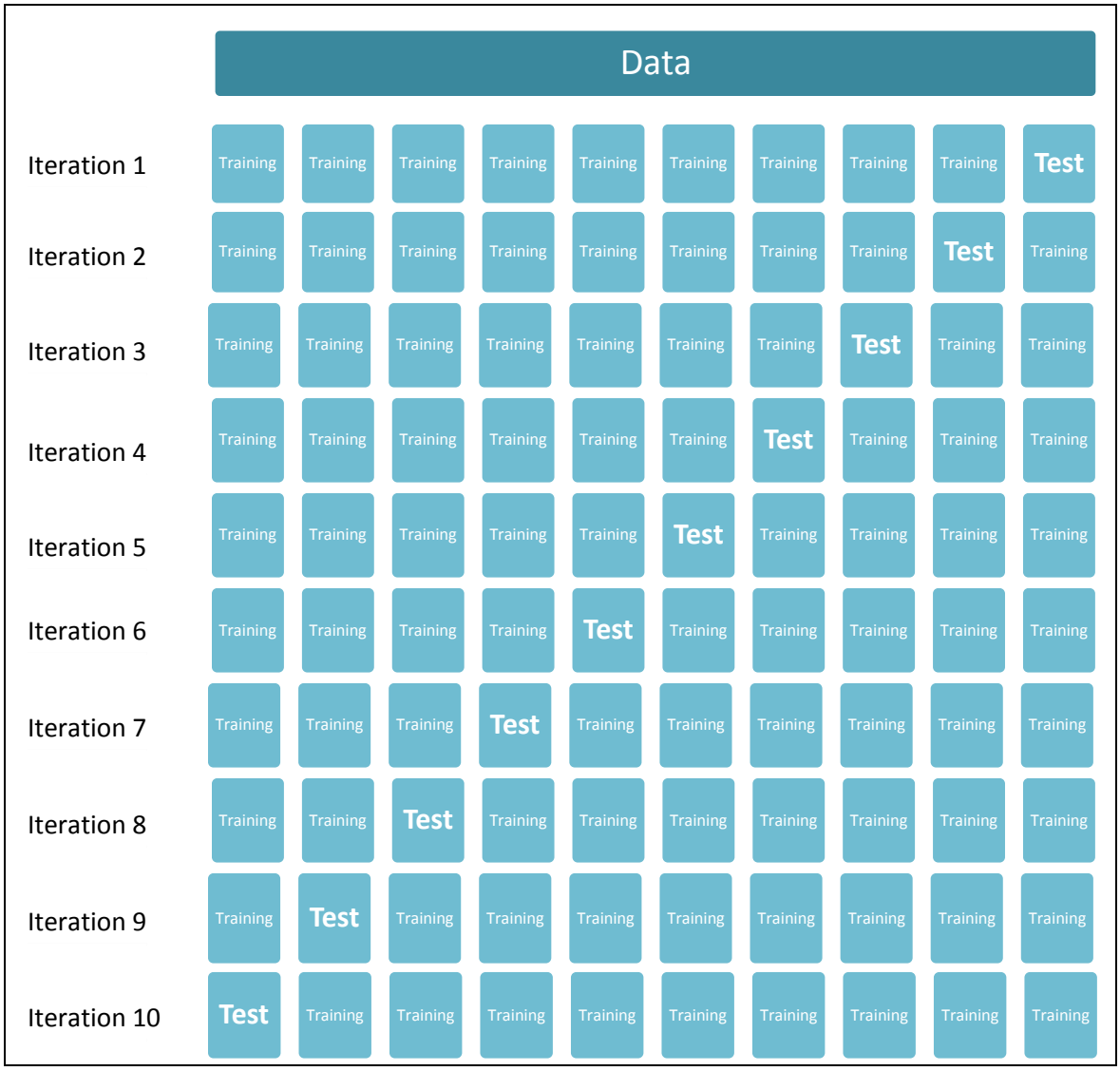


Figure 6.1 10-Fold Cross-Validation Method: Visual Example

	Social Domain			
	Accuracy (%)	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Voter	**71.43**	94.23	66.22	**77.78**
SVM	67.33	96.47	62.35	75.74
NB	54.09	51.29	**80.83**	62.76
DT	52	**100**	52	68.42

Table 6.2 Results of the Accuracy, Recall, Precision, and F-measure of the Social Domain.

	Economic Domain			
	Accuracy (%)	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Voter	**92.22**	87.76	**97.73**	**92.48**
SVM	90.67	**95.33**	88.14	91.59
NB	90.67	86	95.04	90.29
DT	76	60.67	94.44	73.88

Table 6.3 Results of the Accuracy, Recall, Precision, and F-measure of the Economic Domain.

	Education Domain			
	Accuracy (%)	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Voter	**77.91**	**96.81**	73.39	**83.49**
SVM	75	97.71	70.19	81.70
NB	60.48	37.24	**84.91**	51.77
DT	56.80	**100**	56.80	72.45

Table 6.4 Results of the Accuracy, Recall, Precision, and F-measure of the Education Domain.

	Sports Domain			
	Accuracy (%)	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Voter	**94.21**	91.18	**98.41**	**94.66**
SVM	93.57	93.02	95.01	94
NB	92.82	**99**	88.47	93.44
DT	64.63	30.76	96.25	46.62

Table 6.5 Results of the Accuracy, Recall, Precision, and F-measure of the Sports Domain.

	Political Domain			
	Accuracy (%)	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Voter	**88.30**	**90.48**	84.44	**87.36**
SVM	84.70	69.50	**100**	82.01
NB	85.04	70.04	**100**	82.38
DT	48.72	30	47.87	36.88

Table 6.6 Results of the Accuracy, Recall, Precision, and F-measure of the Political Domain.

	All Domain			
	Accuracy (%)	**Recall (%)**	**Precision (%)**	**F-Measure (%)**
Voter	80.08	**97.87**	73.99	84.27
SVM	**81.57**	97.66	75.77	85.33
NB	79.83	71.36	**89.73**	79.50
DT	76.67	67.33	89.26	76.76

Table 6.7 Results of the Accuracy, Recall, Precision, and F-measure of the All Domain.

6.4 Discussion

The goal of the voter model is to take good results from the three models. Figure 6.2 shows how accuracy is better in the voter model than other models. It is enhancing accuracy about **20%** on average. The enhancement appears more clearly in the social and education domains due to the reduction of accuracy in NB and DT.

We found that the SVM has better accuracy than the NB model. This also shown in papers [3] and [51], which are both in the Arabic language. The author in [3] shows how the SVM has better accuracy, about 3.43%, than NB. The SVM shows an improvement of 4-6% over NB in [51].

DT has less accuracy in the five domains due to the small number of tweets (500). When the number of tweets increase (1,500) as in (all domain), the accuracy is good (**76.67**%).

The best accuracies are in the sports and economic domains, as shown in Figure 6.3. This is because of the limited positive and negative words in these domains. For example, in the sports domain, positive words can be (فوز، صدارة), and negative (خسارة، صعوبة). In the economic domain, examples of positive words are (ربح، ارتفاع), and negative words (خسر، انخفاض).

The social domain has the smallest accuracy (**71.43)** due to the plurality of hashtags in it. These hashtags could possibly branch out into more domains, for example (women, man).

Figure 6.4 shows the average of the F-measure in the different domains of Voter, SVM, NB, and DT. The voter model has a better result, **87.15,** than other models. There is an

improvement of up to 25% of F-measure. If we compare our result with [52] by El-Halees, which concludes f-measure ranges of 86.63-88.87%, we have better f-measure ranges (**77.78-94.66**%).

In the all domain, the results are not accurate due to the plurality of domains. The only joint factor is the Arabic language. If we compare our result to the paper [51], which has the same language and work level (twitter), we reached an accuracy of **80.08,** but they have an accuracy of **72.6**. We enhanced the accuracy about **8%**.

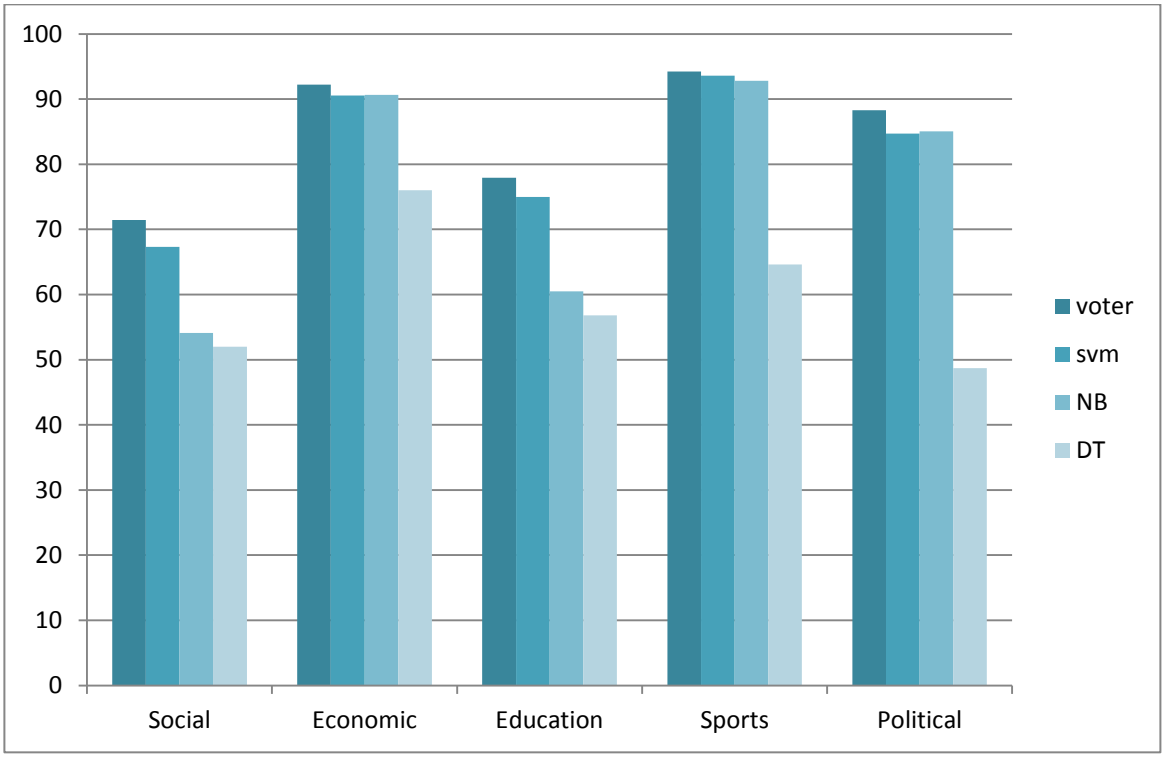


Figure 6.2 Accuracy of Voter, SVM, NB, and DT in Different Domains

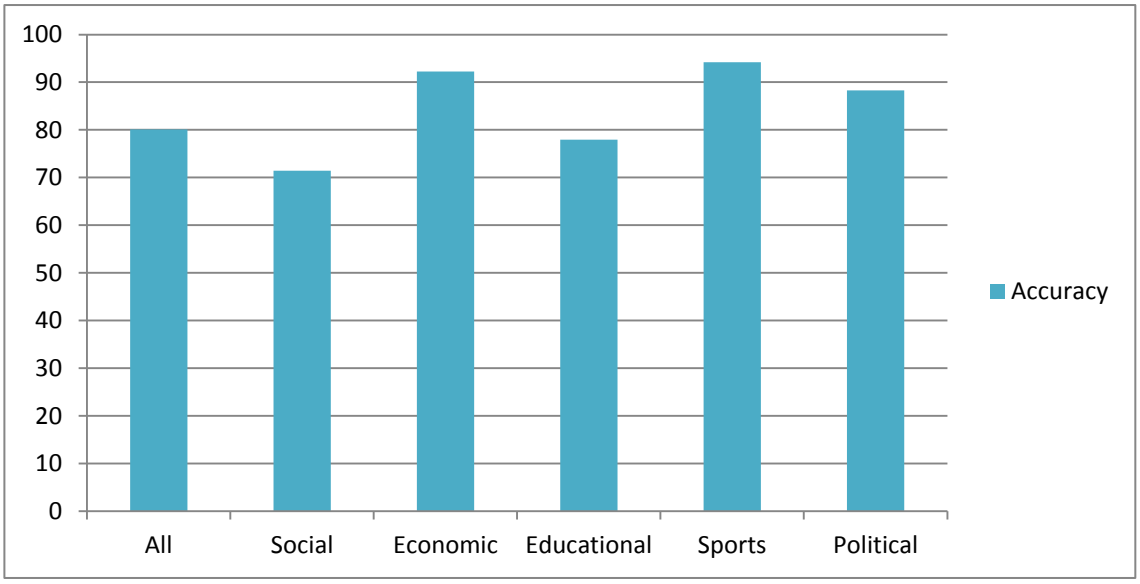


Figure 6.3 Accuracy of Voter Model in Different Domains

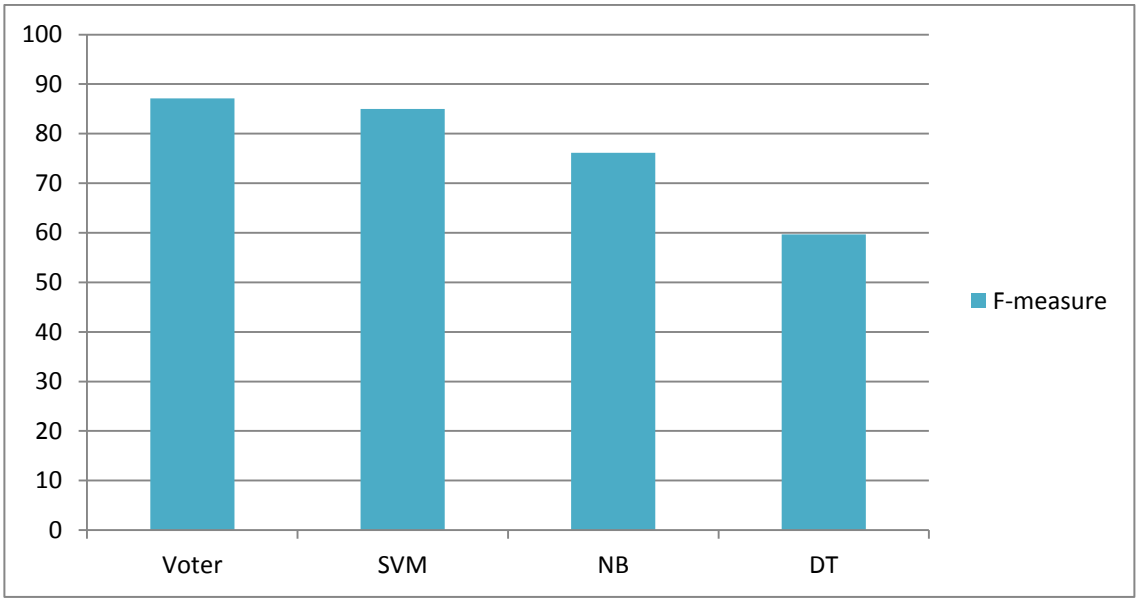


Figure 6.4 Average of F-measure in Different Domains of Voter, SVM, NB, and DT

6.5 Availability of the Voter Model

The availability of the module is the percentage of time when the system is operational. The formula of it is [76]:

$$\text{Availability} = \frac{MTBF}{(MTBF+MTTR)} \qquad (6.5)$$

While MTBF is the mean time between failures, which is the (average) time between failures of a system, and MTTR is the mean time to repair, which is the average time required to repair a failed component or device.

Since we have three algorithms working in parallel and independently, MTTR is zero. If one of the algorithms has any failure, the system can continue the work. So the availability becomes one that means the voter model is available all the time; this is the goal of the fault tolerance technique.

6.6 Impact of Performance on Time

We found good accuracy when using the voter model, but we wanted to find its effect on how long it takes to calculate. Let's assume the three algorithms work sequentially (when the algorithm and the next starts). In this situation, the accuracy is good, but it takes more time to calculate especially if the number of tweets increases. But we ran three algorithms in parallel, so we reached good accuracy with less time. Figure 6.5 shows the comparison between the times of the three algorithms working sequentially and when they work in parallel depending on the number of tweets. We can see that, when the number of tweets increases, the enhancement is more than 50%. As shown in

[77], which is about adding big integer numbers in sequential and parallel algorithms, the parallel algorithm speeds up execution time 320% when four processors are used.

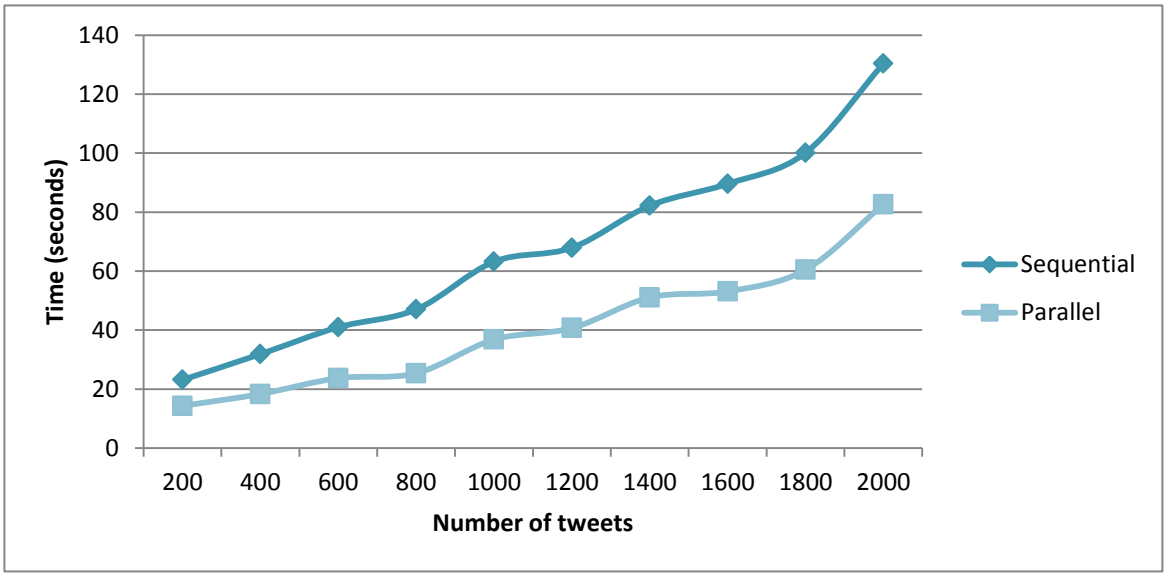


Figure 6.5 Times in sequential and parallel processing depending on the number of tweets

Chapter VII
Conclusion and Future Work

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The existing works on opinion mining in microblogging are limited since the phenomenon has only appeared in the last few years. Most research found is at the document level and deals with the English language. The goal of this thesis is to present a new Arabic corpus for the opinion-mining task in microblogs like Twitter. There were some challenges due to Twitter's data and the Arabic language. To solve this problem, we built a machine learning-based sentiment analysis system for mining and analyzing the Arabic tweets in social networks to determine the positive and negative sentiments. The classification technique used a fault tolerance technique and different machine-learning algorithms (SVM, NB, and DT).

We made the voter model, which is based on the n-version fault tolerance technique in different machine-learning algorithms. Three algorithms working in parallel produce different results. The voter will take the results from three algorithms and compute the final predicted label for the tweet. The tweets collected are in different domains to reach a more accurate result. The domains were social, economic, educational, sports, and political.

We also built an application that determines the percentage of positive and negative opinions based on certain hashtags in specific domain. The application runs on the user selecting the hashtag name and its domain. Those areas were educational, social,

economic, sports, and political. It shows the percentage of positive and negative tweets in addition to the number of tweets written in this hashtag and the time it takes to process the calculation.

The study yielded the following results: the average accuracy of the work based on the voter model is **84.8%**. The best accuracies are in the economic and sports domains (**92.2%** and **94.2**%), and this is due to the limited positive and negative words in these domains. F-measure has ranges (**77.78**-**94.66**) of the voter model in the different domains. The algorithms inside the voter are running in parallel, and when we make a comparison with the sequential run, we found that execution time in parallel was enhanced more than 50%.

The availability of the voter model is equal to **one,** which means the system is available all the time, even if any one of the algorithms fail. By this, we archive to the goal of the fault tolerance technique.

There is some overhead entailed in the voter technique, such as cost and time. The cost overhead comes from the three algorithms used and the voter building. The time overhead is due to running the three algorithms and computing in the voter model.

7.2 Future Work

Many areas for further exploration exist. A good starting point for future research may include adding more domains and specific features for the Arabic language such as POS tagging or a dialect feature, in addition to an analysis of microblogging that includes positive and negative opinions at the same time (complex opinion). Complex opinions

should not be classified as positive or negative only. Moreover, we would use the semantic method to analyze opinions. One of the good examples of using the semantic orientation approach is based on employing one of the lexical resources such as ArabicWordNet (AWN). AWN is an Arabic version of SentiWordNet, which is a lexical resource for opinion mining.

List of References

[1] Alaa El-Halees, "Arabic Opinion Mining Using Combined Classification Approach," in *The International Arab Conference on Information Technology (ACIT)*, Sudan, 2010.

[2] Muhammad Abdul-Mageed, Sandra Kuebler, and Mona Diab, "SAMAR: A System for Subjectivity and Sentiment Analysis of Arabic Social Media," , Jeju, Korea, 2012, pp. 19--28.

[3] Mohammed Rushdi-Saleh, M. Teresa Martín-Valdivia, L. Alfonso Ureña-López, and José M. Perea-Ortega, "OCA: Opinion corpus for Arabic," vol. 62, no. 10, 2011.

[4] Dr. Matthew North, *Data Mining for the Masses*. Washington, USA: Global Text Project, 2012.

[5] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, 2nd ed.: Morgan Kaufmann, 2006.

[6] Osmar R. Zaïane, "Principles of Knowledge Discovery in Databases," University of Alberta, 1999.

[7] David L Olson and Dursun Delen, "Data Mining Process," in *Advanced Data Mining Techniques*.: Springer, 2006.

[8] Release. (2008) Oracle Database. [Online]. http://www.oracle.com/accessibility/

[9] Heikki Mannila and Padhraic Smyth David Hand, *Principles of Data Mining*.: Massachusetts Institute of Technology, 2001.

[10] Andreas Nurnberger,and Gerhard Paaß Andreas Hotho, "A Brief Survey of Text Mining," *Journal for Computational and Language Technology*, pp. 19-62, 2005.

[11] Marti Hearst. (2003) http://people.ischool.berkeley.edu/~hearst/text-mining.html.

[12] J.J.Verbeek, "An information therotic approach to finding word groups for text classification,Master 's thesis," *Institute for Logic, Language and Computation (ILLC)*, 2000.

[13] Brij M. Masand, Stephen J. Smith, and David L. Walt Robert H. Creecy, "Trading MIPS and memory for knowledge engineerring," *Communications of the ACM*, vol. 35, no. 8, pp. 48-64, 1992.

[14] Bhumika, Prof Sukhjit Sing Sehra, and Prof Anand Nayyar, "A Review Paper on Algorithms Used for Text Classification," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 3, pp. 90-99, 2013.

[15] S. KOTSIANTIS,and V. TAMPAKAS M. IKONOMAKIS, "Text Classification Using Machine Learning Techniques," *WSEAS TRANSACTIONS on COMPUTERS*, vol. 4, no. 8, pp. 966-974, 2005.

[16] Ethem Alpaydın, *Introduction to machine learning*, 2nd ed.: Massachusetts Institute of Technology, 2010.

[17] Xindong Wu et al., "Top 10 algorithms in data mining," *Springer*, vol. 14, no. 1, pp. 1-37, january 2008.

[18] Vikramaditya Jakkula, "Tutorial on Support Vector Machine (SVM)," School of EECS,Washington State University, Pullman, 99164,.

[19] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM : a library for support vector machines," National Taiwan University, Taiwan, 2001.

[20] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification," 2003.

[21] Asa Ben-Hur and Jason Weston, "A User's Guide to Support Vector Machines," *Springer*, vol. 609, pp. 223-239, 2010.

[22] B. E. Boser, I. Guyon, and V. Vapnik., "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning*, 1992, pp. 144-152.

[23] Istvan Pilaszy, "Text Categorization and Support Vector Machine," Department of Measurement and Information Systems, Budapest University of Technology and Economics, 2011.

[24] Thorsten Joachims, "Text Categorization with Suport Vector Machines: Learning with Many Relevant Features," *ACM*, pp. 137-142, 1998.

[25] http://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[26] Vidhya.K.A and G.Aghila, "A Survey of Naïve Bayes Machine Learning approach in Text Document Classification," *International Journal of Computer Science and Information Security*, vol. 7, no. 2, 2010.

[27] J. R. Quinlan, "Induction of Decision Trees," *Springer*, vol. 1, no. 1, pp. 81-106, 1986.

[28] Ku TH, Jan RH, Wang K, Tseng YC,and Yang SF. Hu YJ, "Decision tree-based learning to predict patient controlled analgesia consumption and readjustment," in *BMC Med Inform Decis Mak*, Taiwan, 2012, pp. 12-131.

[29] M. Stone, "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111-147, 1974.

[30] Sylvain Arlot and Alain Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40-79, 2010.

[31] Andrew Moore. Statistical Data Mining Tutorials. [Online]. http://www.autonlab.org/tutorials/

[32] Prabir Burman, "A comparative study of ordinary cross-validation,v-fold cross-validation and the repeated learning-testing methods," *Biometrika*, vol. 76, no. 3, pp. 503-514, 1989.

[33] George Forman and Martin Scholz, "Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement," *SIGKDD Explorations*, vol. 12, no. 1, pp. 49-57.

[34] Algirdas Avizienis, "Fault-Tolerant Systems," *IEEE*, vol. 25, no. 12, pp. 1304-1312, 1976.

[35] H. Ammar, B. Cukic, C. Fuhrman, and A. Mili, "A Comparative Analysis of Hardware and Software Fault Tolerance: Impact on Software Reliability Engineering," Institute for Software Research, 1999.

[36] Zaipeng Xie, Hongyu Sun, and Kewal Saluja, "A survey of Software Fault Tolerance Techniques," University of Wisconsin-Madison/Department of Electrical and Computer Engineerin, USA, 2008.

[37] Robert Hanmer, "N-Version Programming," Alcatel-Lucent, 2009.

[38] Danielle Moran and Patrick Dunleavy Amy Mollett, "Using Twitter in university research, teaching and impact activities," LSE Public Policy Group, 2011.

[39] Kevin Makice, *Twitter API: Up and Running*, 1st ed.: O'Reilly Media, 2009.

[40] (2013) Overview: Version 1.1 of the Twitter API. [Online]. https://dev.twitter.com/docs/api/1.1/overview

[41] Michael Sippey. (2012, Augest) Changes coming in Version 1.1 of the Twitter API. [Online]. https://blog.twitter.com/2012/changes-coming-to-twitter-api

[42] Arabic Language. [Online]. http://www.arabic-language.org/

[43] Mohtanick Jamil, "Arabic Words And All Their Glory," *The Caravan Press*, no. 5, 2010.

[44] Mohamed El Kourdi, Amine Bensaid, and Tajje-eddine Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," in *20th International Conference on Computational Linguistics*, Geneva, 2004.

[45] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M.S Khorsheed, and A. Al-Rajeh, "Automatic Arabic Text Classification," *9es Journées internationales d'Analyse statistique des Données Textuelles*, pp. 77-83, 2008.

[46] F. Harrag, E. El-Qawasmeh, and P. Pichappan, "Improving arabic text categorization using decision trees," in *First International Conference on Networked Digital Technologies*, 2009, pp. 110-115.

[47] Mohammed Rushdi-Saleh, M. Teresa Martìn-Valdivia, L. Alfonso Ureña-López, and José M. Perea-Ortega, "Bilingual Experiments with an Arabic-English Corpus for Opinion Mining," , Hissar, Bulgaria, 2011, pp. 740--745.

[48] F. Lazhar and T.G. Yamina, "Identification of opinions in Arabic newspapers," in *International Conference on Machine and Web Intelligence (ICMWI)*, 2010, pp. 317-319.

[49] A.S. Ghareb, A.R. Hamdan, and A.A. and Bakar, "Text associative classification approach for mining Arabic data set," in *Conference on Data Mining and Optimization (DMO)*, 2012, pp. 114-120.

[50] Muhammad Abdul-Mageed, Mohammed Korayem, and David J. Crandall, "Subjectivity and Sentiment Analysis of Arabic: A Survey," in *Advanced Machine Learning Technologies and Applications(AMLTA)*, 2012, pp. 128-139.

[51] Amira Shoukry and Ahmed Rafea, "Sentence-level Arabic sentiment analysis," in *International Conference on Collaboration Technologies and Systems (CTS)*, Denver, CO, USA, 2012, pp. 546 - 550.

[52] Alaa El-Halees, "OPINION MINING FROM ARABIC COMPARATIVE SENTENCES," in *The 13th International Arab Conference on Information Technology*, 2012, pp. 265-271.

[53] Alexander Pak and Patrick Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," in *Proceedings of The Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010.

[54] Alexandra Balahur , Ralf Steinberge, Erik van der Goot, Bruno Pouliquen, and Mijail Kabadjov, "Opinion Mining on Newspaper Quotations," in *International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, vol. 3, 2009, pp. 523-526.

[55] G. Vinodhini and RM. Chandrasekaran, "Sentiment Analysis and Opinion Mining: A Survey," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 6, 2012.

[56] Grigori Sidorov1 et al., "Empirical Study of Machine Learning Based Approach for Opinion Mining in Tweet," in *Advances in Artificial Intelligence*, vol. 7629, 2012, pp. 1-14.

[57] Changli Zhang, Wanli Zuo, Tao Peng, and Fengling He, "Sentiment Classification for Chinese Reviews Using Machine Learning Methods Based on String Kernel," in *Third International Conference on Convergence and Hybrid Information Technology*, vol. 2, 2008, pp. 909-914.

[58] V. Kumar and S. Minz, "Mood Classifiaction of Lyrics using SentiWordNet," in *International Conference on Computer Communication and Informatics (ICCCI)*, 2013, pp. 1-5.

[59] Hassan, Yulan He, and Harith Alani Saif, "Semantic sentiment analysis of twitter," in *The Semantic Web–ISWC*, 2012, pp. 508-524.

[60] Thorsten Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," *Lecture Notes in Computer Science* , vol. 1398, pp. 137-142, 1998.

[61] James Tin-yau Kwok, "Automated Text Categorization Using Support Vector Machine," in *In Proceedings of the International Conference on Neural Information Processing (ICONIP}*, 1998, pp. 347--351.

[62] István Pilászy, "Text Categorization and Support Vector," in *The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*, 2005.

[63] A. Basu, C. Watters, and M. Shepherd, "Support Vector Machines for Text Categorization," in *Proceedings of the 36th Hawaii International Conference on System Sciences*, 2003.

[64] Mahmood H.Kadhim and Nazlia Omar, "Automatic Arabic Text Categorization using Bayesian Learning," in *Computing and Convergence Technology (ICCCT)*, 2012, pp. 415-419.

[65] H. M. Noaman, S. Elmougy, A. Ghoneim, and T. Hamza, "Naive Bayes Classifier based Arabic document categorization," in *Informatics and Systems (INFOS)*, 2010, pp. 1-5.

[66] Madeeh Al-Gedawy and Osman Hegazy, "Handling Text Mining Problems in Arabic using Domain-Specific Approach ," *International Journal of Computer Applications*, vol. 45, no. 16, May 2012.

[67] Rajkumar Buyya, Thamarai Selvi Somasundaram, and Xingchen Chu, *Object Oriented Programming with Java: Essentials and Applications*., 2009.

[68] eclipse. [Online]. http://www.eclipse.org/

[69] Sebastian Land and Simon Fischer, "RapidMiner in Academic Use," 2012.

[70] Markus Hofmann and Ralf Klinkenberg, *"RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*., 2013.

[71] "Integrating RapidMiner into your application," Rapid I, 2012.

[72] (2013) http://rapidminer.com/documentation/.

[73] twitter4j. [Online]. http://twitter4j.org

[74] George Forman and Martin Scholz, "Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 49-57, June 2010.

[75] Max Bramer, "Measuring the Performance of a Classifier," in *Principles of Data Mining*.: Springer, 2007, pp. 173-185.

[76] Jihong Zeng, "A Case Study on Applying ITIL Availability Management Best Practice ," *Contemporary Management Research* , vol. 4, no. 4, pp. 321-332, Dec 2008.

[77] Youssef Bassil and Aziz Barbar, "Sequential & Parallel Algorithms For the Addition of Big-Integer Numbers," *International Journal of Computational Science*, vol. 4, pp. 52-69, 2010.

[78] Ibrahim Abu El-Khair, "Effects of Stop words Elimination for Arabic Information Retrieval: A Comparitive stydy," *international journal of computing and information sciences*, vol. 4, no. 3, pp. 119-133, 2006.

[79] Istvan Pilaszy, "Text Categorization and Support Vector Machine," Department of Measurement and Information Systems, Budapest University of Technology and Economics., 2011.

[80] Dr. Matthew North, *Data Mining for the Masses*.: Global Text Project, 2012.

[81] Barros, R.C. , Basgalupp, M.P., de Carvalho, A.C.P.L.F., and Freitas, A.A., "A Survey of Evolutionary Algorithms for Decision-Tree Induction," *IEEE*, vol. 42, no. 3, pp. 291 - 312, May 2012.