إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Arabic Opinion Mining Using Parallel Decision Trees

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name:	اسم الطالب/ة: وفاء أحمد
Signature:	التوقيع: وفاء
Date:	التاريخ: 2 أغسطس 2015

The Islamic University Of Gaza
Deanery of Graduate Studies
Faculty of Information Technology
Information Technology Department

Arabic Opinion Mining Using Parallel Decision Trees

By:

Wafa A. M. Ahmed

Supervised By:

Dr. Alaa El-Halees

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology

Shaban 1435H - June 2014

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا   هاتف داخلي: 1150

الرقم.ج س. غ/35/......... Ref

التاريخ.........2015/01/24م Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحثة/ **وفاء علاء الدين محمود أحمد** لنيل درجة الماجستير في **كلية *تكنولوجيا المعلومات*** **برنامج تكنولوجيا المعلومات** وموضوعها:

التنقيب عن الآراء العربية باستخدام شجرة القرار المتوازية

Arabic Opining Mining Using Parallel Decision Trees

وبعد المناقشة التي تمت اليوم الثلاثاء 04 ربيع الآخر 1436هـ، الموافق 2015/01/24م الساعة الحادية عشرة صباحاً بمبنى اللحيدان، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. علاء مصطفى الهليس	**مشرفاً ورئيساً**	....
د. رواية فوزي عوض الله	**مناقشاً داخلياً**	Rawia Awad Allah (RA)
د. محمد عبد اللطيف راضي	**مناقشاً خارجياً**	....

وبعد المداولة أوصت اللجنة بمنح الباحثة درجة الماجستير في **كلية *تكنولوجيا المعلومات*/ برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحها هذه الدرجة فإنها توصيها بتقوى الله ولزوم طاعته وأن تسخر علمها في خدمة دينها ووطنها.

والله ولي التوفيق،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

٢٠١٥

أ.د. فؤاد علي العاجز

شئون البحث العلمي والدراسات العليا
الجامعة الإسلامية غزة
The Islamic University
Research & Graduate Affairs

بِسمِ اللهِ الرَّحمنِ الرَّحيمِ

DEDICATION

- ◙ **To my beloved father ....**
- ◙ **To my beloved mother ....**
- ◙ **To my husband ....**
- ◙ **To my girls ....**
- ◙ **To my sisters and brothers ....**
- ◙ **To my family ....**
- ◙ **To my coworkers ....**
- ◙ **To my best friends ....**

Wafa A. M. Ahmed

ACKNOWLEDGEMENTS

Praise is to Allah for giving me the power and help to accomplish this research. Without the grace of Allah, I was not able to accomplish this work.

Many thanks and sincere gratefulness go to my supervisor Dr. Alaa El-Halees, without his valuable guide, assistance, and continuous follow-up; this research would never have been.

In addition, I would like to extend my thanks to the academic staff of the Faculty of Information Technology who helped me during my master's study and taught me different courses.

I also extend my thanks to my parents very much for their pray, patience, motivation, and continues support.

I am greatly indebted to my husband for his support during my course studies and during my thesis work.

Finally yet importantly, I would to thank my family members for their motivation and support.

Wafaa A. M. Ahmed

June 2014

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST Of ABBREVIATIONS

Abbreviation	Stands for
OM	Opinion Mining
KNN	k-Nearest Neighbor
SVM	Support Vector Machine
NB	Naïve Bayes
OC	Opinion Classification
TF	Term Frequency
OCA	Opinion Corpus for Arabic
SSWIL	Slang Sentiment Words and Idioms Lexicon
MSA	Modern Standard Arabic
LVQ	Learning Vector Quantization
SNP	Saudi Newspapers
OSAC	Open Source Arabic Corpus
SMP	Shared Memory multi- Processor
MWK	Moving Window K
ID3	Interactive Dichotomizer version 3
DS	Decision Stump
TF-IDF	Term Frequency - Inverse Document Frequency
TF	Term Frequency
TN	True Negative
TO	Term Occurrences
TP	True Positive
FP	False Positive
FN	False Negative
VSM	Vector Space Model
BHA	Booking of Hotels for Arabic
OCA	Opinion Corpus for Arabic
NLP	Natural Language Processing

ABSTRACT

With the popularity of online shopping it is increasingly becoming important for manufacturers and service providers to ask customers to review their product and associated service. Similarly, the number of customer reviews that a product receives grows rapidly and can be in hundreds or even thousands. This makes it difficult for a potential customer to decide whether or not to buy the product. It is also difficult for the manufacturer of the product to keep track and manage customer opinions. Hence the importance stemmed opinion mining which is an emerging area of research, that summarizes the customer reviews of a product or service and express whether the opinions are positive or negative. Various methods have been proposed as classifiers for opinion mining such as Naïve Bayesian, k-Nearest Neighbor techniques, and Support vector machine, the main drawback of these methods is classifying opinion without giving us the reasons about why the instance opinion is classified to certain class. Therefore, in our work, we investigate opinion mining of Arabic text at the document level, by applying decision trees classification method to have clear, understandable rules. In addition, we apply parallel decision trees classifiers to have efficient results.

We applied parallel decision trees on two Arabic corpus BHA and OCA of text. To generate text representations, we apply some preprocessing operators such as Tokenize , filters Arabic stopwords, Stem Arabic, filters tokens based on their length, and filters tokens based on their content to exclude English words. In case of applying parallel decision tree family on OCA, we get the best results of accuracy (93.83%) , f-measure (93.22) and consumed time 42 Sec at thread 4, which is greater than sequential that have accuracy (92.59%) and f-measure (92.58), and consumed time 68 Sec. In case of applying parallel decision tree family on BHA we get the best results of accuracy (90.63%) , f-measure (82.29)and consumed time 219 Sec at thread 4, these results are different from sequential that have accuracy (90.70%) and f-measure (90.94), and consumed time 417 Sec.

Keywords: Opinion mining, Decision trees, Classification, Arabic text, Parallel Decision Tree, Machine learning, Sentiment Analysis, Sentiment Classification.

الملخص باللغة العربية

بات التسوق عبر الانترنت منتشر بشكل متزايد، مما جعله محط اهتمام الشركات المصنعة ومزودي الخدمات ليسألوا المستهلكين والمراجعين حول منتجاتهم وخدماتهم، وبالمثل آراء المستهلكين والمراجعين حول الخدمات والمنتجات تتزايد لتصبح بالمئات والالاف، هذا التزايد يصعب الامر على المستهلك والمراجع في صنع قراره بشأن الشراء، ومن جانب آخر هو صعب أيضاً على مقدمي الخدمات والشركات المنتجة في تتبع الآراء وإدارتها.

ومن هنا نبعت أهمية مجال تنقيب الآراء الناشئ في مجال البحث وهو الذي يعمد لتلخيص آراء مستهلكي المنتجات ومراجعي الخدمات لتعبر عنهم بالإيجاب والسلب.

هناك العديد من الطرق المقترحة كمصنفات للآراء على سبيل المثال طريقة Naïve Bayes وطريقة k-Nearest Neighbor ، وطريقة Support Vector Machine. هذه الطرق تصنف الآراء إلى إيجابي وسلبي دون إعطائنا تفسير لماذا هذا المثال إيجابي وهذا المثال سلبي. لذلك في بحثنا هذا نصنف الآراء العربية على مستوى المستند بطريقة شجرة صنع القرار لنحصل على تفسير للتصنيفات الإيجابية والسلبية. كما اننا نستخدم أيضاً طرق شجرة صنع القرار المتوازية من أجل الحصول على النتائج بسرعة وكفاءة عالية.

لقد قمنا بتطبيق طرق شجرة صنع القرار على مجموعتين من البيانات المجموعة الأولى تتحدث عن آراء الناس في الفنادق التي تزورها، والمجموعة الثانية تتحدث عن آراء الناس في الأفلام التي يشاهدونها. ومن أجل توليد التمثيلات النصية قمنا باستخدام مشغل الترميز، ومشغل فلتر الكلمات العربية المراد حذفها، ومشغل التجذير العربي، ومشغل تصفية الرموز على أساس طولها، وتم تشغيل فلتر استبعاد الكلمات الإنجليزية.

في حالة تطبيق مجموعة طرق شجرة صنع القرار المتوازية على مجموعة البيانات الخاصة بالأفلام حصلنا على دقة 93.83% وعامل القياس ف 93.22% باستخدام 4 أجهزة وزمن التنفيذ **42** ثانية، اما في حالة التطبيق التسلسلي فكانت الدقة 92.59% وعامل القياس ف 92.58% وزمن التنفيذ 68 ثانية.

في حالة تطبيق مجموعة طرق شجرة صنع القرار المتوازية على مجموعة البيانات الخاصة بالفنادق حصلنا على دقة 90.63% وعامل القياس ف 82.29% باستخدام 4 أجهزة وزمن التنفيذ 219 ثانية، اما في حالة التطبيق التسلسلي فكانت الدقة 90.70% وعامل القياس ف 90.94% وزمن التنفيذ 417 ثانية.

الكلمات المفتاحية:

تنقيب الآراء، شجرة صنع القرار، التصنيف، النصوص العربية، شجرة صنع القرار المتوازية، تدريب الآلة، تحليل المشاعر، تصنيف المشاعر.

Chapter 1

Introduction

In this chapter, we introduce an overview of the thesis, we give a brief description of opinion mining, and classification. In addition, it states the thesis problem, the significance of the thesis, and the scope and limitation of the thesis work.

1.1 Overview

In recent years, there has been a growing interest in the automatic detection of opinionated content in natural language text. Broadly speaking, textual information in the world classified into two main categories, facts and opinions. Facts are true about something and can be tested or proven [1]. Opinions are subjective belief that reflects people's sentiments or perceptions about the entities and events [2].

People's opinion becomes an essential part of our information-collection behavior. In order to make decision, before the Web individual asks for opinions from friends and families, organization is interested in knowing consumer opinions about its products and services; it conducts surveys, opinion polls, consultants, and focused groups. But the world has changed with the user generated content on the Web. This online word-of-mouth behavior represents a major source of information, which is useful to both individuals and organizations. One can post reviews of products at merchant sites and express views on almost anything in reviews, forums, blogs, social networks, micro-blogs, which are collectively called the user generated contents [3][4].

Opinion mining (OM) or sentiment analysis is a recent discipline at the crossroads of information retrieval and computational linguistics, which is concerned not with the topic a document or sentence, is about, but with the opinion, it expresses. The main objective of opinion mining is to extract attributes and components of the object that have been commented on in each document and to determine whether the comments are positive or negative [5].

The large number of opinion mining publication has been published in English, but in Arabic it's still immature and has less number of publications. Opinion mining in Arabic language is very problematic due to the specific morphological and structural changes in the language. First, Arabic grammar is highly complex. Different types of sentence structures can exist in Arabic: verbal, where the sentence starts with a verb phrase, and nominal, where the sentence starts with a noun phrase. Additionally the language allows for different variants within each type of sentence. Many different parts of speech, particular to Arabic, are possible. Furthermore, Arabic is a highly inflectional and derivational language with many word forms and diacritics The same three-letter root can give rise to different words with different meanings. Moreover, the same word can have several different forms with different suffixes, affixes, and prefixes. Special labels called diacritics are used instead of vowels and they differ according to the word form and the part of speech [6].

Opinion mining can be performed at word level, sentence level, or document level. There are several methods that classify opinion to positive or negative ; in our research we will use decision trees.

Opinion mining can be seen as classification problem where review is classified as positive or negative. Classification "is a data mining and machine learning technique used to predict group membership for data instances" [7]. The goal of classification is to accurately predict the target class for each case in the data [8]. To achieve this goal the given data set is split into two disjoint sets training set(seen data) and test set(unseen data), training set used to build the model and test set used to validate it [9].

Classification have many methods e.g. k-nearest neighbors, Naïve bays, neural networks and others, Decision trees is a common method in classifications, It is a hierarchical structure consisting of node and directed edges. It has three types of nodes:

- **A root node** that has not incoming edges and zero or more outgoing edges.
- **Internal nodes**, each of them has exactly one incoming edge and two or more outgoing edges.
- **Leaf or terminal nodes**, each of them has exactly one incoming edge and no outgoing edges.

In a decision tree, each node is assigned a class label. The root and other nodes contain attribute test conditions to separate records that have different characteristics [9].

Some researchers used decision trees to investigate the impact of text preprocessing, our work will use decision trees classification method for the following reasons:

- It doesn't depend on a parameter value of k such as k-nearest neighbor (KNN) [10].
- It doesn't need conditional probability such as Naïve Bayesian prediction which requires each conditional probability be non-zero. Otherwise, the predicted probability will be zero [11].
- An easily-understandable model. When looking at a decision tree, it is easy to see that some initial variable divides the data into two categories and then other variables split the resulting child groups. This information is very useful to the researcher who is trying to understand the underlying nature of the data being analyzed; a neural network is more of a “black box” that delivers results without an explanation of how the results were derived. Thus, it is difficult or impossible to explain how decisions were made based on the output of the network [12].
- When it is trained, they can be expressed in rule-based manner such as rule based classification method [13].
- Decision trees generate rules. A rule is a conditional statement that can easily be understood by humans.
- Decision trees generate rules that are easily used within a document to identify text related to rule.

In several applications mainly including data mining, the dataset to be learned is very large. In those cases, it is highly desirable to construct univariate decision trees in reasonable time. This may be accomplished by parallelizing univariate decision tree algorithms [14].

1.2 Problem statement

The huge amount of opinions in the documents with high dimensionality and in particular in the Arabic language which has a rich nature and very complex morphology require a large amount of computational power for classification. In addition, most classifiers classify opinion without giving us the reasons about why the instance opinion is classified to certain class. Therefore, the problems this research tries to solve are how to effectively handle Arabic opinion mining to obtain more efficient and understandable rules.

1.3 Objectives

1.3.1 Main Objective

The main objective of this research is, to effectively handle Arabic opinion mining by using parallel decision trees to classify documents as positive or negative and to get more understandable rules and efficient results.

1.3.2 Specific Objectives

The specific objectives of this research are:

- Examine the current approaches on opinion mining and Arabic opinion mining; determine the problem in their classification methods in order to be avoided in our research.
- Find appropriate corpus suitable for our problem.
- Investigating the most suitable text preprocessing techniques such as stemming and term pruning methods and term weighting schemes.
- Apply a family of sequential decision trees classifiers to Arabic opinion documents.
- Apply a family of parallel decision trees classifier to the same Arabic opinion documents in previous step.
- Compare results between the sequential and parallel classifiers, to know what is better to obtain more efficient and understandable rules.
- Compare decision trees classification results with other classifiers such as with rule induction, support vector machine (SVM), k-Nearest Neighbor (k-NN), and Naïve Bayes classification methods along the four criteria, which are accuracy, recall, precision, and f-measure.
- Evaluating the family of parallel decision trees classifiers using different performance metrics for parallel systems such as execution time, speedup, efficiency, and parallel overhead.

1.4 Scope and Limitation

- We use existing opinion mining preprocessing methods that can applied to Arabic documents such as String Tokenization, Stop words Removal, Arabic Stemming Algorithm, and Term Pruning.

- We apply the opinion preprocessing techniques using the open source machine-learning tool RapidMiner.
- Our work is limited to classify opinions in Arabic opinion reviews only. We will not include classifying opinion reviews in other languages, such as: English and European in the same review.
- Our work will concentrate on documents level because Arab tell us their opinion in more details.
- We classify opinions using the open source machine-learning tool RapidMiner to apply a family of sequential and parallel decision trees.
- For applying parallel decision trees, we will conduct our experiments on a set of threads, The maximum number of the threads will be subject to the experiment.
- We will use two Arabic corpus; the first one is OCA from http://sinai.ujaen.es/wiki/index.php/OCA_Corpus_(English_version), and the second is BHA which is collected from online Arabic economic websites, including tripadvisor.com.eg , booking.com, and agoda.ae

1.5 Significance of the thesis

- The growth of participation in the Internet fortifies the importance of public opinion as well as the use of public polls for different topics that many websites already employ. Most customers express their opinions on various kinds of entities, such as products and services. These reviews not only provide customers with useful information for reference, but also are valuable for merchants to get the feedback from customers and enhance the qualities of their products or services. Therefore, mining opinions from these vast amounts of reviews becomes urgent, and has attracted many attentions from many researchers.
- Saving efforts and time by helping the producer and consumers, such as: commerce to get the feedback from consumers and enhance the qualities of their products or services, also to give consumer knowledge about best products in Arabic opinion reviews quickly.
- OM research published in the Arabic language is very little, and need to intensify efforts.
- The OM published research tells us about the number of positive and negative opinions but do not tell us why we get these numbers, this is why we need to use DT.
- Performance of most OM methods are not efficient so we plan to use parallel methods.
- More support for the Arabic language in the technology area as Islam encourages us to support it.

1.6 Research Methodology

To accomplish the objectives of the research, the following methodology will be followed (see Figure 1.1):

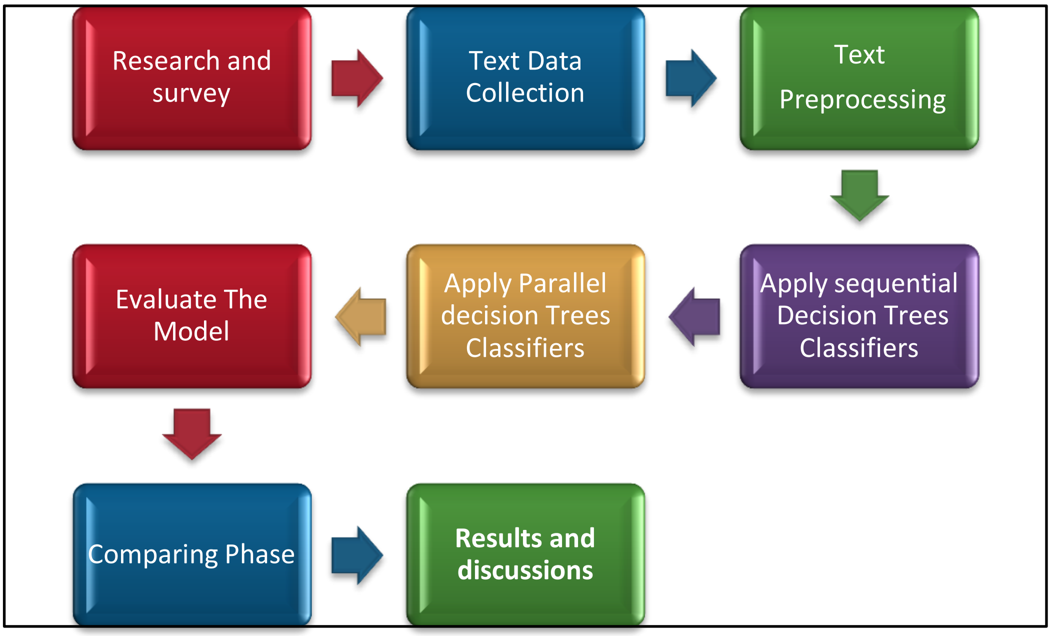


Figure 1.1: Methodology Steps

1.6.1 Research and survey

This include reviewing the recent literature closely related to the thesis problem statement and the research question. After analyzing the existing methods, identifying the drawbacks or the lack of existing approaches. We formulate the strategies and solutions and how to overcome the drawbacks.

1.6.2 Text Data Collection

We collect two various corpus for design experimental Arabic corpus, the first one is OCA from http://sinai.ujaen.es/wiki/index.php/OCA_Corpus_(English_version), and the second is BHA which is collected from online Arabic economic websites, including tripadvisor.com.eg , booking.com, and agoda.ae .

1.6.3 Text Preprocessing

Text preprocessing is the important stage in text classification, It includes tokenizing strings to words, normalizing the tokenized words, applying stop words removal, applying the suitable term stemming and pruning methods as a feature reduction techniques, and finally applying the suitable term weighting scheme to enhance text document representation as feature vector. We use the open source machine-learning tool RapidMiner for text preprocessing.

1.6.4 Apply Sequential Decision Tree classifiers

We apply a family of decision trees learning algorithms such as decision tree, Quinlan's ID3 numerical, and decision stumps on two corpus.

1.6.5 Apply Parallel Decision Tree classifiers

We apply a family of Parallel decision trees learning algorithms such as parallel decision tree, parallel Quinlan's ID3 numerical, and parallel decision stumps on two corpus.

1.6.6 Evaluate The Model

Evaluating the family of parallel decision trees classifiers using different performance metrics for parallel systems such as execution time, speedup, efficiency, and parallel overhead. And evaluating the obtained classification results using different classification measures such as accuracy, precision, recall, and F-measure.

1.6.7 Comparing Phase

We compare results between the sequential and parallel classifiers, and compare decision trees classification results with other classifiers such as with rule induction, support vector machine (SVM), k-Nearest Neighbor (k-NN), and Naïve Bayes classification methods along the four criteria, which are accuracy, recall, precision, and f-measure

1.6.8 Results and discussions

In this stage, we present and analyze the experimented results after learning OCA and BHA corpus by the sequential and parallel decision trees learning algorithms. After analyzing, we will justify parallel decision trees feasibility.

1.7 Research Format

This thesis consists of six mainly chapters, which are structured around the objectives of the research. The main points discussed throughout the chapters are listed below:

- ***Chapter 1 Introduction:*** It gives an overview of the thesis; first, it gives a brief description of opinion mining, and classification. In addition, it states the thesis problem, the research objectives, the significance of the thesis, the scope and limitation of the thesis work, and the research methodology.
- ***Chapter 2 Related Works:*** It presents other works related to the thesis and will discuss the state of art and literature survey.
- ***Chapter 3 Arabic Opinion Mining And Classifications:*** It describes Arabic Language, discusses the complexity of Arabic Language, introduces opinion mining (OM) , opinion classification (OC), presents an overview of decision tree family, performance metrics for opinion mining, discusses the need for parallel decision trees, parallel computing, performance metrics for parallel computing, and finally presents an overview of parallel decision trees.
- ***Chapter 4 Text Data Collection And Preprocessing:*** It describes the Arabic text data collection that is collected from various resources, and apply preprocessing stages

including feature reduction using morphological analysis techniques, and term weighting.

- ***Chapter 5 Experimental Results and Evaluation:*** It gives in detail about the sets of experiments, and analyzes the experimental results. In addition, it gives a discussion for each set experiment. Then, it produces some experiments to comparison goals.

Chapter 6 Conclusions and Future Work: It discusses the conclusions and presents possible future works.

Chapter 2

Related works

Many researchers have worked on opinion classification in English and other European languages such as French, German, and Spanish and in Asian languages such as Chinese and Japanese [15 , 16, and 17]. However, researches on opinion mining for Arabic language are limited [23, 24, 25, and 26].

In this chapter, we will discuss works done in the area of our research, which is using decision tree in Arabic opinion mining. Research published in the field of opinion mining in Arabic language is still few and need to increase the focus research. In this section, we review the most important of them.

We reviewed works related to our researches on the field of Arabic opinion classification into three categories: Investigating the impact of Arabic opinion preprocessing, applying sequential classification algorithms on Arabic opinion, and applying parallel classification algorithms on Arabic opinion.

2.1 Investigating the Impact of Arabic Opinion Preprocessing

The researchers used some preprocessing methods that has an impact in Arabic opinion mining, the following are the most important researches:

Farra et.al. In [20] investigated a novel grammatical approach, which overcomes the limitations of multiple Arabic sentence structures by considering a general structure for the Arabic sentence. They investigated the semantic approach, which is based on the semantic orientation of words, and their corresponding frequencies so that they built an interactive learning semantic dictionary, which stores the polarities of the roots of different words and identifies new polarities based on these roots. For document-level classification, they used sentences of known classes to classify whole documents, using a novel approach whereby documents are divided dynamically into chunks and classification is based on the semantic contributions of different chunks in the document. This dynamic chunking approach can also be investigated for sentiment mining in other languages. Finally, they proposed a hierarchical classification scheme that uses the results of the sentence-level classifier as input to the document level classifier.

Said et al. In [21] provided an evaluation study of several morphological tools for Arabic Text Categorization using SVMs. Their study includes using the raw text, the stemmed text, and the root text. The stemmed and root text are obtained using two different preprocessing tools. The results revealed that using light stemmer combined with a good performing feature selection 30 method such as mutual information or information gain enhances the performance of Arabic Text Classification for small sized datasets and small threshold values for large datasets. Additionally, using the raw text leads to the worst performance in small datasets while its performance was among the best tools in large datasets. This may explain the contradiction in the results obtained previously in the literature of the Arabic text categorization since the performance of the preprocessing tools is affected by the characteristics of the dataset used.

Duwairi et.al. In **[**22**],** compared three dimensionality reduction techniques which are: stemming, light stemming, and word cluster. Stemming reduces words to their stems.

Light stemming removes common affixes from words without reducing them to their stems. Word clusters group synonymous words into clusters and each cluster is represented by a single word. The purpose of employing the previous methods is to reduce the size of documents vectors without affecting the accuracy of the classifiers. They used k-NN to perform the comparison. The comparison metric includes size of documents vectors, classification time, and accuracy (in terms of precision and recall). They used Term Frequency (TF) as a weighting scheme for feature selection. Several experiments were carried out using four different representations of the same corpus: the first version uses stem-vectors, the second uses light stem-vectors, the third uses word clusters, and the fourth uses the original words (without any transformation) as representatives of documents. In terms of vector sizes and classification time, the stemmed vectors consumed the smallest size and the least time necessary to classify a testing dataset.. The light stemmed vectors superseded the other three representations in terms of classification accuracy. The feature selection and reduction strategies can decrease the computation complexity, reduce the dimensionality, and improve the accuracy rate of classification. However, this approach could not do well in the case of reducing computation complexity for text documents with high number of distinct words and in particular in the Arabic language which has a rich nature and very complex morphology. In addition, this approach reduces the features but what is the solution in the case of large volume of text documents which increase the computation complexity.

Saad in [18] presented and compared the impact of text preprocessing, which has not been addressed before, on Arabic text classification using popular text classification algorithms; Decision Tree, *K* Nearest Neighbors, Support Vector Machines, Naïve Bayes and its variations. Text preprocessing includes applying different term weighting schemes, and Arabic morphological analysis (stemming and light stemming). He implemented and integrated text classification algorithms applied on seven Arabic corpora (3 in-house collected and 4 existing corpora). From his experimental results, he showed the following conclusions:

First, They cannot avoid feature reduction for Arabic language to reduce complexity for classifiers, he concluded that light stemming and term pruning is the best feature reduction technique because light stemming is more proper than stemming from linguistics and semantic view point, and it has the least preprocessing time, it also has superior average classification accuracy. Second, Support Vector Machines SVMs is a robust classifier even in high dimensions. Language consideration in Naïve Bayes NB variants improved performance. SVMs and NB variant have superior performance and achieved the best classification accuracy. Third, Term weighting schemes have direct impact on distance based classifiers. Distance based classifiers also affected by the used distance metric.

Saleh et.al. in [19] presented a new Arabic corpus for the opinion mining task that has been made available to the scientific community for research purposes. To generate the Opinion Corpus for Arabic (OCA) they have extracted the reviews from different web pages about movies. OCA comprises 500 reviews in Arabic, of which 250 are considered as positive reviews and the other 250 as negative opinions. That process involved collecting reviews from several Arabic blog sites and web pages using a simple bash script for crawling. Then, they removed HTML tags and special characters, and spelling mistakes were corrected manually.

Next, a processing of each review was carried out, which involved tokenizing, removing Arabic stop words, and stemming and filtering those tokens whose length was less than two characters. Specifically, they have used the. In their experiments, they have used only the basic Arabic stemmer of Rapid Miner and the Arabic stop word list provided by the same software. Finally, three different n-gram schemes are generated (unigrams, bigrams, and trigrams) and cross validation is applied to evaluate the corpus. They compared their work with support vector machines and Naïve Bayes; they found that results obtained by their method are very promising

2.2 Applying Sequential Classification Algorithms on Arabic Opinion Mining

El-Halees In **[23]** founds that using one method on Arabic opinioned documents produce a poor performance. So, he used a combined approach that consists of three methods. First method is lexicon based which is used to classify as much documents as possible. The second method is maximum entropy which used the resultant classified documents from first method as training. Then maximum entropy produces accurate results if they can classify the document, using another classifier. The third method is k-nearest which used the classified documents from lexicon based method and maximum entropy as training set and classifies the rest of the documents. He applied his method on 1143 posts contains 8793 Arabic statements; his system achieved an accuracy of 80.29%. The accuracy almost went from 50% using one method, 60% using two method and 80% using three methods, which is a satisfactory performance especially for complex language such as Arabic. The experimental results further show that recall and precision of positive documents are better than the negative one.

Soliman1et al. in [27] proposed an opinion mining approach to mine unstructured and ungrammatical customers' Arabic comments based in new Slang Sentiment Words and Idioms Lexicon (SSWIL). The new lexicon collected manually from news websites, Facebook and Twitter pages, which were used as interaction and communication pages between web users. SVM technique was applied with SSWIL to classify comments to satisfy or dissatisfy comments. The classifier consists of three main phases: Arabic comments, data preparation, Data preprocessing, and data classification. They worked on users comments and SSWIL enhances the classification task to be 86.86% of classified comments instead of 75.35% when using classical opinion words lexicon with precision 88.63 and recall 78 instead of 82.4 and 59.33 respectively.

Abdul-Mageed et.al. in [28] presented a newly labeled corpus of Modern Standard Arabic(MSA) from the news domain manually annotated for subjectivity and domain at the sentence level. They summarized their linguistically motivated annotation guidelines and provided examples from their corpus exemplifying the different phenomena. Throughout their paper, they discussed expression of subjectivity in natural language, combining various previously scattered insights belonging to many branches of linguistics.

Almas et.al. in [29] proposed a pattern discovery algorithm to facilitate opinion mining in that they have captured the essence of financial news. It is typically news about change, for example, the token percent plays a key role in English, Arabic and Urdu. Their method can generate value judgment about whether a sentence contains a positive sentiment or a negative sentiment or both. In a similar vein, their method does not aggregate the sentiment over many sentences that comprise a news report to generate a sentiment index of the report. However, their method shows that the principal keyword collocates in a statistically significant manner with metaphorical and literal words for the direction of change. Moreover, that one can generate an aggregated (direction of change) index for a news story and indeed for a collection of texts.

To illustrate how the algorithm work they show the results obtained from running the algorithm on two comparable English and Arabic financial corpora of 2.75 million tokens and a smaller1.03 million tokens Urdu financial corpus.

They investigated the effect of preprocessing the Arabic corpora and particularly collapsing clitics, and how their method can utilize other properties in the distribution of words in the financial corpora for extracting features for the automatic classification (clustering) of words and patterns as positive and negative, particularly: (a) the word order of each language (e.g. Arabic lead sentences start with a verb that is predominantly a reporting or a movement verb and weirdness analysis can filter the movement verbs) and the relationship between words in the titles and lead sentences (b) the preliminary observation that positive news is more abundant than negative news (asymmetry) across the three languages.

Elhawary et al. in [30] demonstrated a system for mining Arabic business reviews from the web. The system comprises two main components: they a reviews classifier that classifies any webpage whether it contains reviews or not by using AdaBoost classifier, and a sentiment analyzer that identifies the review text itself and identifies the individual sentences that actually contain a sentiment (positive, negative, neutral or mixed) about the business being reviewed . They provide their users the information they need about the local businesses in the language they understand, and therefore provided a better search experience for the Middle East region, which mostly speaks Arabic. The system is of particular interest for languages that are of poor web content, e.g., Arabic; and can easily be extended to other alike languages.

2.3 Arabic Classification

Opinion mining used text classification methods. The following are examples of researches in Arabic text classification

Harrag et al. in [24] improved Arabic text classification by feature selection based on hybrid approach. He used decision tree algorithm and reported classification accuracy of 93% for scientific corpus, and 91% for literary corpus. He collected 2 corpora; the first one is from the scientific encyclopedia “Do You Know“ (هل تعلم). It contains 373 documents belonging to 1 of 8 categories (innovations, geography, sport, famous men, religious, history, human body, and cosmology), each category has 35 documents. The second corpus is collected from Hadith encyclopedia (موسوعة الحديث الشريف) from the nine books (الكتب التسعة). It contains 435 documents belonging to 14 categories.

Kheirsat in [25] used N-grams frequency statistics to classify Arabic text, she addressed high dimensional text data by mapping text documents to set of real numbers representing tri-grams frequency profile. The N-gram method is language independent and works well in the case of noisy-text. She classified a test text document by computing Manhattan/Dice distance similarity measure to all training documents and assign the class of the training document with smallest/largest computed distance to the test text document. She reported that Dice outperforms Manhattan distance measure. Although the Manhattan measure has provided good classification results for English text documents, it does not seem to be suitable for Arabic text documents. She collected her corpus from Jordanian newspapers (Al-Arab, Al-Ghad, Al-Ra'I, Ad-Dostor). The corpus belongs to 1 of 4 categories (sport, economic, weather, and technology). She applied stop words removal and used 40% for training and 60% for testing.

El-Halees et al. in [31] introduced Arabic text classification through Learning Vector Quantization LVQ algorithm. They used different versions of LVQ (LVQ2.1, LVQ3, OLVQ1 and OLVQ3) algorithms, to determine which LVQ versions has higher accuracy and less time. They selected Arabic documents from different domains. After that, they selected suitable pre-processing methods such as term weighting schemes, and Arabic morphological analysis (stemming and light stemming), these preprocessing prepared dataset that need for classification. The first results show that LVQ2.1 has highest accuracy (93.08) compared to other LVQ's algorithms. Also, LVQ2.1 achieved approximately (94%) when using light stemming as Arabic morphological, tf-idf term weighting techniques and term frequency=5. Finally, LVQ2.1 as neural network algorithm is able to obtain a high accuracy in less time.

Duwairi in [32] compared the performance of three classifiers for Arabic text categorization. She used NB, KNN, and distance-based classifiers. Unclassified documents were preprocessed by removing punctuation marks and stop words. Each document is then represented as a vector of words (or of words and their frequencies as in the case of the naïve Bayes classifier). Stemming was used to reduce the dimensionality of feature vectors of documents. The accuracy of the classifiers was measured using recall, precision, fallout and error rate. The three classifiers were tested using in-house collected Arabic text. Unclassified documents were categorized using the three classifiers in turn. The results showed that the performance of the Naïve Bays classifier outperformed the other two classifiers.

Alsaleem in [33] discussed the problem of automatically classifying Arabic text documents. They used the NB algorithm, which is based on probabilistic framework, and Support Vector Machine algorithm SVM algorithm to handle their classification problem. The average of three measures obtained against Saudi Newspapers (SNP) Arabic data sets indicated that the SVM algorithm outperformed NB algorithm regards to F1, Recall and Precision measures.

2.4 Parallel Classification Algorithms

Since we proposed to use parallel decision tree in opinion mining, which is an application of text classification, we list the following works, which used parallel algorithms for classification:

Abu Tair in [26] developed a parallel classifier for large-scale Arabic text that enhanced the level of speedup, scalability, and accuracy. The proposed parallel classifier based on the sequential k-NN algorithm. He tested the parallel classifier using the Open Source Arabic Corpus (OSAC) that includes 22,428 text documents. Each text document belongs to 1 of 10.He experimented the parallel classifier on a multicomputer cluster that consists of 14 computers. The experimental results on the performance indicate that the parallel classifier design has very good speedup characteristics when the problem sizes are scaled up. In addition, classification results showed that the proposed classifier has achieved accuracy, precision, recall, and F-measure with higher than 95%.

Zaki et al. in [34] presented parallel algorithms for building decision tree classifiers on shared-memory multiprocessor (SMP) systems. The proposed algorithms span the gamut of data and task parallelism. The Moving-Window-K (MWK) algorithm uses data parallelism from multiple attributes, but also uses task pipelining to overlap different computing phase within a tree node, thus avoiding potential sequential bottleneck for the hash-probe construction for the split phase. The MWK algorithm employs conditional variable, not barrier, among leaf nodes to avoid unnecessary processor blocking time at a barrier. It also exploits dynamic assignment of attribute files to a fixed set of physical files, which maximizes the number of concurrent accesses to disk without file interference. The SUBTREE algorithm uses recursive divide-and-conquer to minimize processor interaction, and assigns “free processors” dynamically to “busy groups” to achieve load balancing. Their experiments show that both algorithms achieve good speedups in building the classifier on a 4-processor SMP with disk configuration and on an 8-processor SMP with memory configuration, for various numbers of attributes, various numbers of example tuples of input databases, and various complexities of data models. The performance of both algorithms are comparable, but MWK overall has a slight edge. These experiments demonstrate that the important data-mining task of classification can be effectively parallelized on SMP machines.

2.5 Conclusion

From previous discussion, works have not been used parallel decision trees on opinion mining, For text classification only Saad in [18] used decision tree to investigate the impact of text preprocessing. In general , publications in Arabic opinion mining field are very little and need increasing efforts, the previous works were on Arabic text classification but my work on opinion mining.

Also, most of related work in the literature used small corpus, only Mohammed Abu Tair in [26] used OSAC which is the largest freely public Arabic corpus of text documents, using one corpus is not enough to evaluate his proposed parallel classifier of KNN.

In this research, we applied a family of parallel decision trees classifiers on two corpus. We investigated the impact, the benefits of using different Arabic morphological techniques with different weighting schemes applied on two corpora, and using three classifiers, which are parallel decision tree C4.5, parallel Quinlan's ID3 numerical, and parallel decision stumps. Moreover, provide a comprehensive study for Arabic opinion classification on booth corpus.

In the next chapter, we provide description about Arabic opinion mining and classifiers that used in this research.

Chapter 3

Arabic Opinion Mining and Classification

This chapter describes Arabic language and its complexity, introduces opinion mining (OM) , opinion classification (OC), presents an overview of decision tree family, performance metrics for opinion mining, discusses the need for parallel decision trees, parallel computing, performance metrics for parallel computing, finally presents an overview of parallel decision trees.

3.1 Arabic Language

Arabic Language is the fifth widely used languages in the world. More than 422 million people speak it as a first language and by 250 million as a second language; Arabic Language belongs to the Semitic language family. Semitic languages are commonly written without the vowel marks, which would indicate the short vowels. Semitic languages can get away with this because they all have a predictable root pattern system [35]. Arabic alphabet consists of the following 28 letters (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 ه و ي) in addition to the Hamza (ء). Arabic letters have different styles when appearing in a word depending on the letter position (beginning, middle or end of a word) and on whether the letter can be connected to its neighbor letters or not. Diacritics are signals placed below or above letters to double the letter in pronunciation or to act as a short vowel. Arabic diacritics include Shada, dama, fatha, kasra, sukon, double dama, double fatha, double kasra. Different letter styles and diacritics make parsing Arabic text a non-trivial task. There is no upper or lower case for Arabic letters like English letters. The letters (أ و ي) are vowels, the rest are constants. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right to left [18,35, 36, 37, 38, 39, 40].

Arabic is a challenging language for a number of reasons:

1. Orthographic (الاملاء) with diacritics is less ambiguous and more phonetic in Arabic, certain combinations of characters can be written in different ways [18].
2. Arabic language has short vowels, which give different pronunciation. Grammatically they are required but omitted in written Arabic texts [41].
3. Arabic has a very complex morphology, as compared to English language Synonyms are widespread. Arabic is a highly inflectional and derivational language [42].
4. Automatic text classification depends on the contents of documents, a huge number of features or keywords can be found in Arabic texts such as morphemes that may be generated from one root, which may lead to a poor performance in terms of both accuracy and time [18, 41, 42].
5. Lack of publicly freely accessible Arabic Corpora [18, 41, 42].

3.2 Opinion Mining

Opinion mining, also called sentiment analysis, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes. It represents a large problem space. There are also many names and slightly different tasks, e.g., sentiment analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining, etc. However, they are now all under the umbrella of sentiment analysis or opinion mining. While in industry, the term sentiment analysis is more commonly used, but in academia, both sentiment analysis and opinion mining are frequently employed. They represent the same field of study. [2, 3, 4 and 43].

The main objective of opinion mining is to extract attributes and components of the object that have been commented on in each document and to determine whether the comments are positive, or negative [5].

Opinions are central to almost all human activities because they are key influencers of our behaviors. Whenever we need to make a decision or we want to know others' opinions. In the real world, businesses and organizations always want to find consumer or public opinions about their products and services. Individual consumers also want to know the opinions of existing users of a product before purchasing it, and others' opinions about political candidates before making a voting decision in a political election. In the past, when an individual needed opinions, he/she asked friends and family. When an organization or a business needed public or consumer opinions, it conducted surveys, opinion polls, and focus groups. Acquiring public and consumer opinions has long been a huge business itself for marketing, public relations, and political campaign companies [43, 44].

With the explosive growth of social media (e.g., reviews, forum discussions, blogs, micro-blogs, Twitter, comments, and postings in social network sites) on the Web, individuals and organizations are increasingly using the content in these media for decision-making. Nowadays, if one wants to buy a consumer product, one is no longer limited to asking one's friends and family for opinions because there are many user reviews and discussions in public forums on the Web about the product. For an organization, it may no longer be necessary to conduct surveys, opinion polls, and focus groups in order to gather public opinions because there is an abundance of such information publicly available. However, finding and monitoring opinion sites on the Web and distilling the information contained in them remains a formidable task because of the proliferation of diverse sites. Each site typically contains a huge volume of opinion text that is not always easily deciphered in long blogs and forum postings. The average human reader will have difficulty identifying relevant sites, extracting, and summarizing the opinions in them. Automated sentiment analysis systems are thus needed. In recent years, we have witnessed that opinionated postings in social media have helped reshape businesses, and sway public sentiments and emotions, which have profoundly influenced our social and political systems [3, 4].

Opinion mining discovers opinioned knowledge at three levels which are:

Document level

The task at this level is to classify whether a whole opinion document expresses a positive or negative sentiment .For example, given a product review, the system determines whether the review expresses an overall positive or negative opinion about the product. This task is commonly known as document-level sentiment classification. This level of analysis assumes that each document expresses opinions on a single entity (e.g., a single product). Thus, it is not applicable to documents which evaluate or compare multiple entities [45,46].

Sentence level:

The task at this level goes to the sentences and determines whether each sentence expressed positive, or negative, or opinion. This level of analysis is closely related to subjectivity classification, which distinguishes sentences (called objective sentences) that express information from sentences (called subjective sentences) that express subjective views and opinions. However, we should note that subjectivity is not equivalent to sentiment as many objective sentences can imply opinions [47, 48].

Entity and Aspect level:

Both the document level and the sentence level analysiss do not discover what exactly people liked and did not like. Aspect level performs finer-grained analysis. Aspect level was earlier called feature level (feature-based opinion mining and summarization) [49]. Instead of looking at language constructs (documents, paragraphs, sentences, clauses or phrases), aspect level directly looks at the opinion itself. It is based on the idea that an opinion consists of a sentiment (positive or negative) and a target (of opinion). An opinion without its target being identified is of limited use. Realizing the importance of opinion targets also helps us understand the sentiment analysis problem better. For example, although the sentence "although the service is not that great, I still love this restaurant" clearly has a positive tone, we cannot say that this sentence is entirely positive. In fact, the sentence is positive about the restaurant (emphasized), but negative about its service (not emphasized). In many applications, entities and/or their different aspects describe opinion targets. Thus, the goal of this level of analysis is to discover sentiments on entities and/or their aspects.

In addition to the previous classification and to make things even more interesting and challenging, there are other types of opinions, i.e., regular opinions and comparative opinions. A regular opinion expresses a sentiment only on a particular entity or an aspect of the entity, e.g., "Coke tastes very good," which expresses a positive sentiment on the aspect taste of Coke. A comparative opinion compares multiple entities based on some of their shared aspects, e.g., "Coke tastes better than Pepsi," which compares Coke and Pepsi based on their tastes (an aspect) and expresses a preference for Coke [3, 49, 50].

The large number of opinion mining publications is in English, but in Arabic, it is still immature and has less number of publications. Opinion mining in Arabic language is very problematic due to the specific morphological and structural changes in the language.

In my work, I will use document level, which is the most common in this area. Since Arabs express their opinion in more details.

3.3 Opinion Classification

Opinion classification has been widely studied by the natural language processing community and is defined as follows: Given a set of text data D, it analyzes whether each document d ∈ D expresses a positive or negative opinion on a specific object [45, 51]. For example, given a set of reviews on movie reviews, the system classifies them into positive reviews and negative reviews. This is almost similar to a supervised classification method but different from the regular topic based text classification, which classifies documents into predefined topic classes, e.g., sports, art etc. In topic-based classification, topic related words are important. However, in opinion classification, topic-related words are not very important but, opinion words that indicate positive or negative opinions are important, e.g., great, excellent, amazing, horrible, bad, worst, etc. Most of the methodologies for opinion mining apply some forms of machine learning techniques for classification. Customized-algorithms specifically for opinion classification have also been developed, which exploit opinion words and phrases together with some scoring functions [46].

3.4 Serial Classification Algorithms

Classification is a supervised technique with labeled examples for the class attribute, which is used as the training set by the classification algorithm, and the unlabeled example for the class attribute, which needs to be found using, the multiple predictor attributes available. Classification accuracy depends on the model being built using the historical data that accurately predicts the label (class) of the unlabeled examples [52]. Popular techniques include Bayesian approach, Decision tree induction approach, Support Vector Machine and Neural network approach.

In our work we use decision trees, Decision trees is a common method in classification, it is a hierarchical structure consisting of nodes and arcs which connect nodes. To make a decision, one starts at the root node, and asks questions to determine which arc to follow, until one reaches a leaf node and the decision is made. The basic structure is shown in Figure 3.1 [53, 54, and 55].

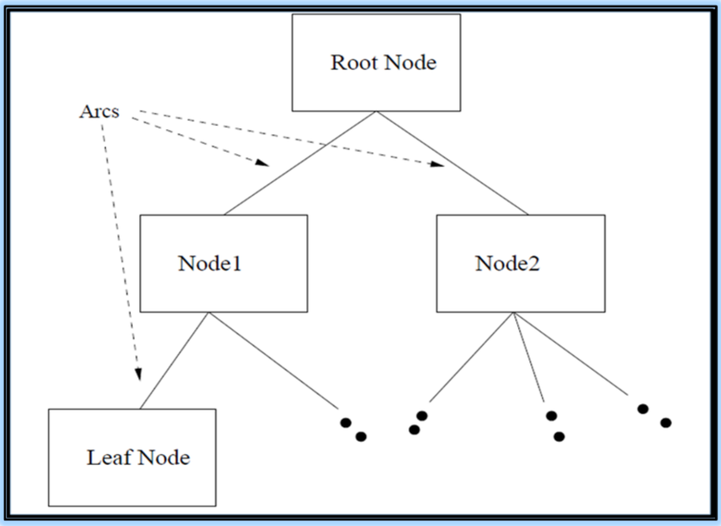


Figure 3. 1: Basic Decision Tree Structure

There is a family of decision trees such as Decision tree, Quinlan's ID3, and Decision stumps classifier.

3.4.1 ID3 Classifier:

In decision tree learning, ID3 is an algorithm invented by Quinlan [54], where "ID" stands for "Interactive Dichotomizer" and "3" stand for "version 3" is a rooted tree containing nodes and edges. Each internal node is a test node and corresponds to an attribute. The edges going out of a node correspond to the possible values of that attribute. The ID3 algorithm works as follows. The tree is constructed top-down in a recursive fashion. At the root, each attribute is tested to determine how well it alone classifies the samples. The "best" attribute is then chosen and the samples are partitioned according to this attribute. The ID3 algorithm is then recursively called for each child of this node, using the corresponding subset of data [53, 54, and 56].

The main ideas behind the ID3 algorithm are :

- ❖ Each non-leaf node of a decision tree corresponds to an input attribute, and each arc to a possible value of that attribute. A leaf node corresponds to the expected value of the output attribute when the input attributes are described by the path from the root node to that leaf node [57].
- ❖ In a “good” decision tree, each non-leaf node should correspond to the input attribute which is the *most informative* about the output attribute amongst all the input attributes not yet considered in the path from the root node to that node. This is because we would like to predict the output attribute using the smallest possible number of questions on average [59].
- ❖ *Entropy* is used to determine how informative a particular input attribute is about the output attribute for a subset of the training data. Entropy is a measure of uncertainty in communication systems introduced by Shannon [58]. It is fundamental in modern information theory.

The ID3 metrics are:

- **Entropy**:
  Entropy $\mathrm{H}(\mathrm{S})$ is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S) [57, 58, and 59].

$$\mathrm{H}(\mathrm{S}) = -\sum_{x \in X} \mathrm{p}(\mathrm{x}) \log_{s} \mathrm{p}(\mathrm{x}) \quad (3.1)$$

Where,

- ❖ S- The current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
- ❖ X- Set of classes in S
- ❖ P(x) The proportion of the number of elements in class x to the number of elements in set S

When H(S)=0, the set S is perfectly classified (i.e. all elements in S are of the same class).

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set Son this iteration. The higher the entropy, the higher the potential to improve the classification here.

Information Gain

Information gain $\mathrm{IG}(\mathrm{A})$ is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in S was reduced after splitting set Son attribute$A$ [57, 58, and 59].

$$\mathrm{IG}(\mathrm{A}) = \mathrm{H}(\mathrm{S}) - \sum_{t \in T} p(t)\,\mathrm{H}(t) \quad (3.2)$$

Where,

- $\mathrm{H}(\mathrm{S})$- Entropy of set S
- T - The subsets created from splitting set S by attribute A such that

$$\mathrm{S} = \bigcup_{t \in T} t \quad (3.3)$$

- $p(t)$ The proportion of the number of elements in $t$to the number of elements in set S.
- $\mathrm{H}(t)$ Entropy of subset t.

Figure 3.2 show the **ID3 Pseudo code [58]:**

```
ID3 (Examples, Target_Attribute, Attributes)
  Create a root node for the tree
  If all examples are positive, Return the single-node tree Root, with label = +.
  If all examples are negative, Return the single-node tree Root, with label = -.
  If number of predicting attributes is empty, then return the single node tree Root,
  With label = most common value of the target attribute in the examples.
  Otherwise Begin
    A ← The Attribute that best classifies examples.
    Decision Tree attribute for Root = A.
    For each possible value, $v_i$, of A,
      Add a new tree branch below Root, corresponding to the test A = $v_i$.
      Let Examples( $v_i$ ) be the subset of examples that have the value $v_i$ for A
      If Examples ( $v_i$ ) is empty
        Then below this new branch add a leaf node with label = most common target
value in the examples
      Else below this new branch add the subtree ID3 (Examples ( $v_i$ ),
Target_Attribute, Attributes – {A})
  End
  Return Root
```

Figure 3. 2: The ID3 Algorithm

3.4.2 C4.5 Decision tree Classifier

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan [60]. C4.5 is an extension of Quinlan's earlier ID3 algorithm [58]. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier [60].It builds decision tree from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set S = s1, s2, s3,…. Of already classified samples. Each sample Si consists of a p-dimensional vector (x1, i, x2,i, …, xp,i), where the xi represent attributes or features of the sample, as well as the class in which Si falls [61, 60]. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recursive on the smaller sub lists [61, 60].

The following is the algorithm, which has a few base cases[60]:

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class
- None of the features provides any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class
- Instance of previously unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value

Improvements from ID3 algorithm

C4.5 made a number of improvements to ID3. Some of these are:

- Handling both continuous and discrete attributes–- In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it [61].
- Handling training data with missing attribute values C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations [61].
- Handling attributes with differing costs [61].
- Pruning trees after creation–- C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes [61].

Figure 3.3 show the C4.5 Pseudo code [60]:

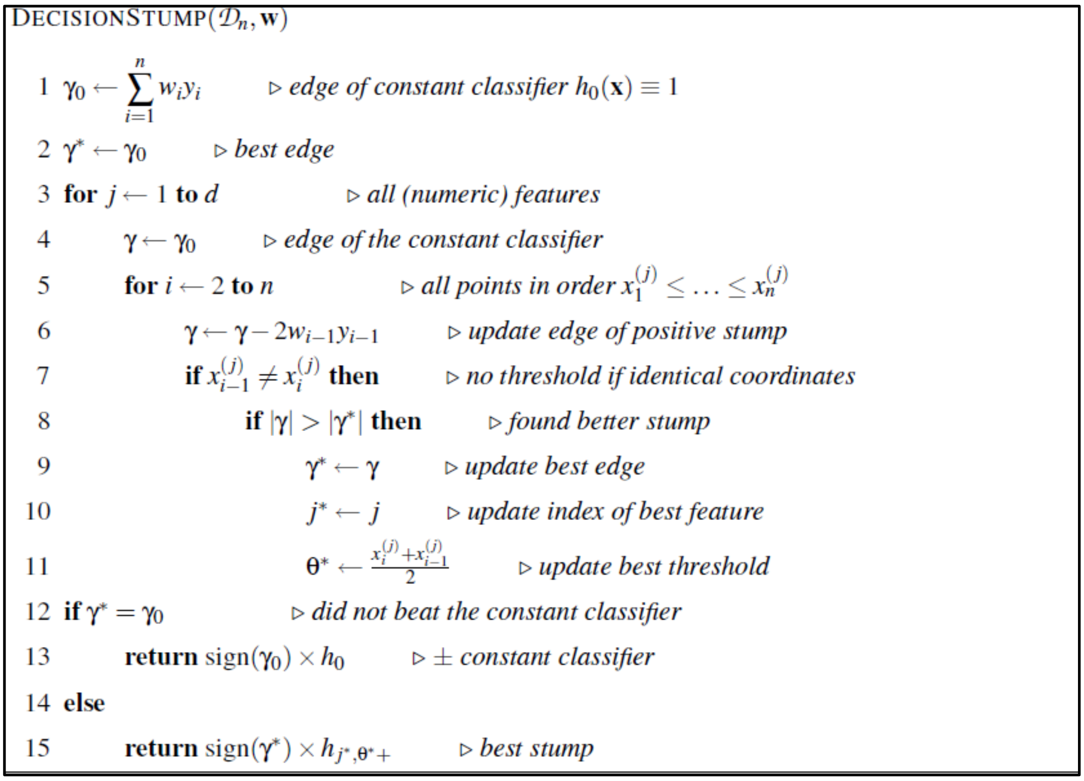
```
Input: an attribute valued dataset D
Tree={}
If D is “pure” OR other stopping criteria met then
        Termenate
End if
For all attribute a ∈ D do
        Compute information-theoritic cretiria if we split on a
End for
$a_{best}$ = Best attribute according to above computed cretiria
Tree = cretae a decision node that tests $a_{best}$ in the root
$D_v$ = induced sub-dataset from D based $a_{best}$
For all $D_v$ do
        $Tree_v$ = C4.5($D_v$)
        Attach $Tree_v$ to the corresponding branch of Tree
End for
Return Tree
```

Figure 3. 3: C4.5 Algorithm

3.4.3 Decision Stump Classifier

A decision stump (DS) is a machine-learning model consisting of a one-level decision tree [44, 62]. That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. Sometimes they also called 1-rules [63, 62]. Depending on the type of the input feature, several variations are possible. For nominal features, one may build a stump, which contains a leaf for each possible feature value, or a stump with the two leaves, one of which corresponds to some chosen category, and the other leaf to all the other categories. For binary features, these two schemes are identical. A missing value may be treated as a yet another category. For continuous features, usually, some threshold feature value is selected, and the stump contains two leaves — for values below and above the threshold. However, rarely, multiple thresholds may be chosen and the stump therefore contains three or more leaves [64].

figure 3.4 show the Decision stump Pseudo code [3]:



```
DECISIONSTUMP($\mathcal{D}_n$, $\mathbf{w}$)
$\gamma_0 \leftarrow \sum_{i=1}^{n} w_i y_i$        ▷ edge of constant classifier $h_0(\mathbf{x}) \equiv 1$
$\gamma^* \leftarrow \gamma_0$        ▷ best edge
for $j \leftarrow 1$ to $d$        ▷ all (numeric) features
    $\gamma \leftarrow \gamma_0$        ▷ edge of the constant classifier
    for $i \leftarrow 2$ to $n$        ▷ all points in order $x_1^{(j)} \leq \ldots \leq x_n^{(j)}$
        $\gamma \leftarrow \gamma - 2 w_{i-1} y_{i-1}$        ▷ update edge of positive stump
        if $x_{i-1}^{(j)} \neq x_i^{(j)}$ then        ▷ no threshold if identical coordinates
            if $|\gamma| > |\gamma^*|$ then        ▷ found better stump
                $\gamma^* \leftarrow \gamma$        ▷ update best edge
                $j^* \leftarrow j$        ▷ update index of best feature
                $\theta^* \leftarrow \frac{x_i^{(j)} + x_{i-1}^{(j)}}{2}$        ▷ update best threshold
if $\gamma^* = \gamma_0$        ▷ did not beat the constant classifier
    return sign$(\gamma_0) \times h_0$        ▷ ± constant classifier
else
    return sign$(\gamma^*) \times h_{j^*,\theta^*+}$        ▷ best stump
```

Figure 3. 4: The Decision Stump Algorithm

3.5 The need for parallel Classification

Decision trees are simple yet effective classification algorithms. One of their main advantages is that they provide human-readable rules of classification. Decision trees have several drawbacks, one of which is the need to sort all numerical attributes in order to decide where to split a node. This becomes costly in terms of running time and memory size, especially when decision trees are trained on large data, and we need to classify it in shorter times this make the classification algorithm an ideal candidate for parallelization. The parallel formulation, however, must address the issues of efficiency and scalability in both memory requirements and parallel runtime. Parallel decision trees overcome the sorting obstacle by applying pre-sorting, distributed sorting, and approximations [65, 66].

3.6 Parallel Computing

Parallel computing is the simultaneous execution of the same task on multiple processors in order to obtain faster results. It is widely accepted that parallel computing is a branch of distributed computing, and puts the emphasis on generating large computing power by employing multiple processing entities simultaneously for a single computation task. These multiple processing entities can be a multiprocessor system, which consists of multiple processors in a single machine connected by bus or switch networks, or a multicomputer system, which consists of several independent computers interconnected by telecommunication networks or computer networks [67].

The main purpose of doing parallel computing is to solve problems faster or to solve larger problems. Parallel computing is widely used to reduce the computation time for complex tasks. Many industrial and scientific research and practice involve complex large-scale computation, which without parallel computers would take years and even tens of years to compute. It is more than desirable to have the results available as soon as possible, and for many applications, late results often imply useless results [26].

3.7 Parallel Decision Tree Algorithms

3.7.1 Parallel Quinlan's ID3 numerical classification algorithm

This method learns decision trees without pruning using both nominal and numerical attributes. Decision trees are powerful classification methods, which often can also easily be understood. This decision tree learner works similar to Quinlan's ID3. This implementation might distribute the work over several threads for utilizing the today's multicore CPUs [59].

3.7.2 Parallel Decision tree classification algorithm

This method learns decision trees from both nominal and numerical data. Decision trees are powerful classification methods that often can also easily be understood. This decision tree learner works similar to Quinlan's C4.5 or CART. This implementation might distribute the work over several threads for utilizing the today's multicore CPUs. The actual type of the tree is determined by the criterion that specifies the used criterion for selecting attributes and numerical splits [55].

parallel implementation of the C4.5 decision tree construction algorithm. This implementation follows a hybrid parallelism strategy with the use of data parallelism at the beginning of the decision tree build process and task parallelism at the lower nodes of the tree which cover a smaller amount of examples.

3.7.3 Parallel Decision stumps classification algorithm

This operator learns decision stumps from both numerical and nominal attributes, the resulted model consisting of a one-level decision tree, That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. Sometimes they are also called 1-rules. This implementation might distribute the work over several threads for utilizing the today's multicore CPUs [62].

3.8 Evaluation

3.8.1 Measuring Effectiveness of opinion mining

In this section we discuss the effectiveness of OM and efficiency of OM. The measures of evaluating the performance of classification are a confusion matrix, which is also called a performance vector that contains information about realistic and predicted classifications.

Table 3.1: confusion matrix table

Predicted			
True		**Positive**	**Negative**
	Positive	(TP) True Positive	(FN) False Negative
	Negative	(FP) False Positive	(TN) True Negative

The entries in the confusion matrix are [68, 69]**:**

- The number of **correct** predictions that an instance is **positive** (TP).
- The number of **correct** predictions that an instance is **negative** (TN).
- The number of **incorrect** predictions that an instance is **positive** (FP).
- The number of **incorrect** predictions that an instance is **negative** (FN).

From the entries in the confusion matrix several concepts have been computed. These concepts will be used in later chapters to evaluate the performance of Appling decision trees classifiers on Arabic opinions. These include Recall, Precision, F-Measure, and accuracy.

1) Accuracy:

The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using this equation [69].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\% \qquad (3.4)$$

2) Recall:

True positive rate, Recall, or Sensitivity which is the proportion of Real Positive cases that are correctly predicted positive. This measures the Coverage of the Real Positive cases by the (Predicted Positive) rule. Recall is defined, with its various common appellations, by equation [69].

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (3.5)$$

3) Precision:

True False Accuracy, Precision or Confidence (as it is called in Data Mining) denotes the proportion of Predicted Positive cases that are correctly Real Positives. This is what Machine Learning, Data Mining and Information Retrieval focus on, Precision is defined, with its various common appellations, by equation [69]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (3.6)$$

4) F-Measure:

F-Measure or F-Factor is the ratio between **recall** and **precision** measurements F-Measure is defined, with its various common appellations, by equation [69].

$$\text{F - Measure} = \frac{2 \times \text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}} \qquad (3.7)$$

3.8.2 Efficiency of OM

In order to demonstrate the efficiency of parallel processing for a problem on some platform, several concepts have been defined. These include serial runtime, parallel runtime, parallel overhead, speedup, and efficiency.

1) Serial Runtime

The serial runtime of a program is the time elapsed between the beginning and the end of its execution on a sequential computer. The serial runtime is denoted by T_S [70].

2) Parallel Runtime

The parallel runtime is the time that elapses from the moment the first processor starts to the moment the last processor finishes execution. The parallel runtime is denoted by T_P [70].

3) Total Parallel Overhead

The parallel overhead is the total time spent by all processors combined in non-useful work [70]. The overhead function (T_o) is given by:

$$T_o = (p\, T_P - T_S) / T_S \qquad (3.8)$$

Where p is the number of processors, T_S is the serial runtime, and T_P is the parallel runtime.

4) Speedup:

The speedup is the ratio of the time taken to solve a problem on a single processor to the time required to solve the same problem on a parallel computer with *p* identical processing elements [56]. This is shown as:

$$S = T_S / T_P \qquad (3.9)$$

where *S* is the speedup achieved with *p* processors, T_S is the serial runtime, and T_P is the parallel runtime. As the number of processors increases, speedup also increases until a saturation point is reached. Beyond this point, adding more processors will not bring further performance gain. This is the combined result of reduced computation on participating node, and increased duplicate computation and synchronization and communication overhead [70].

5) Efficiency

The efficiency is a measure of the fraction of time for which a processing element is usefully employed [56]. It is given by:

$$E = S / p \qquad (3.10)$$

where *E is* the efficiency, *S* is the speedup achieved with *p* processors, and *p* is the number of processors. It measures how much speedup is brought per additional processor. Based on the typical speedup curve shown in Figure 3.5, it is evident that typically efficiency will be decreased upon increase in the number of processors. Efficiency can be as low as 0 and as high as 1 [70].

6) Scalability

The concept of scalability cannot be computed but evaluated. A parallel system is said to be scalable when the efficiency can be kept constant as the number of processing elements is increased, provided that the problem size is increased [70].

3.9 Summary

In this chapter, we presented an overview of Arabic language that is a challenging language; we described opinions that are central to almost all human activities, the main objective of opinion mining is to extract attributes and components of the object and classify them to positive, or negative. We presented a family of decision tree such as ID3, decision stump, and decision tree C4.5. We discussed the reasons for using decision trees. In addition, we referred that the measures of evaluating the performance of classification are a confusion matrix, also we presented an overview of parallel computing, and discussed the need for parallel decision trees, we described the performance metrics for parallel systems that evaluate the effectiveness of parallel programs, and finally we described the parallel decision trees classifiers.

In the next chapter, we will describe the Arabic text data collection and text preprocessing stages.

Chapter 4

Text Data Collection and Preprocessing

a Collection and Preprocessing

This chapter is organized into five sections, Section 4.1, will give a description about text data collection for designing experimental data. Section 4.2, will be about preprocessing stages. Sections 4.3, will be about text preprocessing tool, section 4.4 talk about BHA preprocessing, section 4.5 talk about OCA preprocessing.

To apply and evaluate text classifier, various steps have to be performed. The main required steps are shown in Figure 4.1: In this chapter, we will apply the first two stages and the other stages will be applied in the next chapter. In this chapter we describe the Arabic text data collection which is collected from various resources, these collections to be classified need to have high quality text. The high quality in text mining usually refers to some combinations of relevance, novelty, and interestingness [71]. For those reasons, we apply preprocessing stages including feature reduction using morphological analysis techniques, and term weighting.

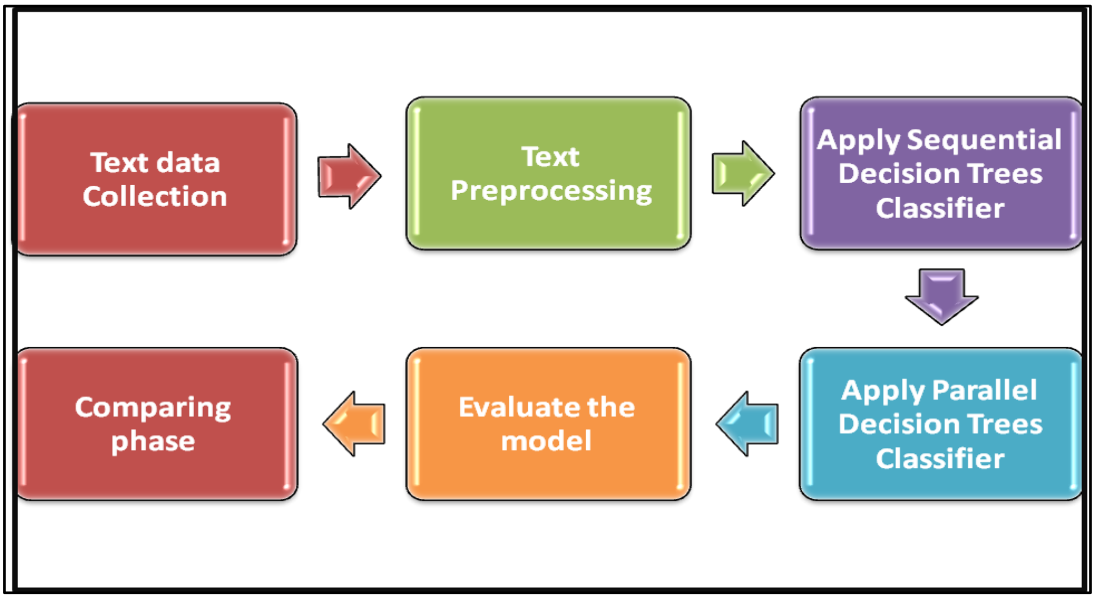

Text data Collection
Text Preprocessing
Apply Sequential Decision Trees Classifier
Apply Parallel Decision Trees Classifier
Evaluate the model
Comparing phase

Figure 4. 1: Methodology Steps

4.1 Arabic Text Data Collection

The first step is text data collection. Data collection is an important step of our work as the inaccurate data collection can impact the result of a study and ultimately lead to invalid results and decisions [72]. We collected two Arabic corpus of text documents which are shown in table 4.1.

Table 4.1: General Information about two Arabic corpus

Arabic Corpus	No of Examples	No of Positive	No of Negative
Booking of hotels (BHA)	8224	4112	4112
Opinion corpus for Arabic (OCA)	500	250	250

4.1.1 Booking of hotels for Arabic (BHA)

The data set Booking of hotels for Arabic (BHA) is collected from online Arabic economic websites, including tripadvisor.com.eg , booking.com, and agoda.ae which has reviews about hotels, resorts, flights, vacation, travel packages, and lots more. With different characteristics and sizes by crawls. These reviews is labeled by users whose write their opinion on the site. We randomly collected the records available from June 2007 to December 2013. The crude reviews included a number of challenges we attempted to fix manually, including filtering out spurious and unrelated comments. We used hotelName and reviewBody attributes. Also we attempted to program a java program to merge the hotelName and reviewBody properties to be in new named text document. The corpus consists of 8224 Text Files, half negative and half positive as shown in table 4.2.

Table 4.2: General Information about BHA Corpus.

Corpus	#Text Files
tripadvisor	2176
Booking	3867
agoda	2181

4.1.2 Opinion corpus for Arabic (OCA)

Opinion corpus for Arabic (OCA) is a corpus of text from movie review sites by Saleh et al. [19]. It consists of 500 reviews, half negative and half positive. The crude reviews included a number of challenges which the authors attempted to fix manually, including filtering out spurious and unrelated comments, Romanization of Arabic, multi-language reviews, and differing spellings of proper names. This corpus is freely available at [73].

4.2 Arabic Text Preprocessing

The second step is text preprocessing. The Arabic language is highly derivative where tens or even hundreds of words could be formed using only one root [74]. Furthermore, a single word may be derived from multiple roots; the language consists of three types of words, nouns, verbs and particles. Nouns and verbs are derived from a limited set of about 10,000 roots [74].Templates are applied to the roots in order to derive nouns and verbs by removing letters, adding letters, or including infixes. Furthermore, a stem may accept prefixes and/or suffixes in order to form the word [75].Text preprocessing is the important stage in text classification and it includes many steps including, String Tokenization, Stopwords Removal, Morphological Analysis Techniques, Term Pruning, and Vector Space Model (VSM) and Term Weighting Schemes.

4.2.1 String Tokenization

String Tokenization is the process of splitting the text of a document into a sequence of pieces called tokens, to specify the splitting points there are several options non letters, specify characters, and regular expression. The first option non-letter character is the default setting. This will result in tokens consisting of one single word, which is the most appropriate option before finally building the word vector. The second option specify characters is appropriate to build windows of tokens or split complete sentences, The third option lets you define regular expressions and is the most flexible for very special cases [76]. we will use the first option which is the most common and effective form.

4.2.2 Stopwords Removal

Stop-words are common words that do not have so much meaning in a retrieval system , do not contribute to the semantics of the documents and have no real added value [77].There is no confirmed list of stop words which all Natural language processing (NLP) tools incorporate. Not all NLP tools use a stopwords. Some tools specifically avoid using them to support phrase searching, For Arabic, stopwords list includes punctuations (? ! …), pronouns (... هو الذي التي هما), adverbs (تحت فوق), days of week (السبت الأحد الإثنين), month of year (..... ابريل فبراير مارس). [78].We will use a common Arabic stopword list.

4.2.3 Morphological Analysis Techniques

For Arabic language, Arabic words are formed from abstract forms named roots, the root is the basic form of word from which many derivations can be obtained by attaching certain affixes. So we produce many nouns and verbs and adjectives from the same root [79]. A root based stemmer main goal is to extract the basic form for any given word by performing morphological analysis for the word [80], Table 4.3 shows an example root "لعب" and a set (not all) derivations can be obtained from this root:

Table 4.3: Some Derivations of the root "لعب"

يلعب	ملعب	لاعب	ملعوب	لعبة
Play	playground	Player	Played	game

There are two different morphological analysis techniques; stemming and light stemming. The term stemming refers to the reduction of words to their roots [77]. Stemming has a large effect on Arabic information retrieval, at least in part due to the highly inflected nature of the language.

1. Arabic Root Stemming Algorithm

Stemming would reduce the Arabic words such as (المكتبة الكاتب الكتاب) which mean (the library), (the writer), and (the book) respectively, to one stem (كتب), which means (write). Khoja’s stemmer removes the longest suffix and the longest prefix. It then matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words[81]. Algorithm steps of Khoja Arabic stemmer is described in Figure 4.2 [82].

1. Remove diacritics
2. Remove stopwords, punctuation, and numbers.
3. Remove definite article (ال)
4. Remove inseparable conjunction (و)
5. Remove suffixes
6. Remove prefixes
7. Match result against a list of patterns.
    1. If a match is found, extract the characters in the pattern representing the root.
    2. Match the extracted root against a list known —valid|| roots
8. Replace weak letters واي with و
9. Replace all occurrences of *Hamza* إ ء ئ with ا
10. Two letter roots are checked to see if they should contain a double character. If so, the character is added to the root.

Figure 4. 2: Arabic Stemming Algorithm Steps

2. Arabic Light Stemming Algorithm

Light stemming, in contrast, removes common affixes from words without reducing them to their stems. For example, stemming would reduce the Arabic words (الكاتب الكتاب المكتبة) which mean (the library), (the writer), and (the book) respectively, to one stem (كتب), which means (write).

The main idea for using light stemming is that many word variants do not have similar meanings or semantics. However, these word variants are generated from the same root. Thus, root extraction algorithms affect the meanings of words. Light stemming aims to enhance the classification performance while retaining the words meanings. It removes some defined prefixes and suffixes from the word instead of extracting the original root**.**Formally speaking, the aforementioned Arabic words (المكتبة الكاتب الكتاب) which mean (the library), (the writer), and (the book) respectively, belong to one stem (كتب) despite they have different meanings. Thus, the stemming approach reduces their semantics. The light stemming approach, on the other hand, maps the word (الكتاب) which means (the book) to (كتاب) which means (book), and stems the word (الكاتب) which means (the writers) to (كاتب) which means (writer). Another example for light stemming is the words (المسافرين المسافرون) which mapped to word (مسافر). Light stemming keeps the words' meanings unaffected. that there are many words morphology have different meaning despite they have the same root. Figure 4.3 shows the steps of Arabic light stemming. Arabic light stemmer from Apache Lucene is standard Arabic light stemmer [82].

1. Normalize word

   – Remove diacritics

   – Replace إ أ آ with ا

   – Replace ة with ه

   – Replace ى with ي

   – Remove diacritics

2. Stem prefixes

   – Remove Prefixes: ، لل ،فال،و،وال،ال،كال،بال

3. Stem suffixes

   – Remove Suffixes: ي ، ه ، ية ، ينٌ ، ون ، ات ، ان ، ها

Figure 4. 3: Arabic Light Stemming Algorithm Steps

We will use Arabic Stemming algorithm as it gives us better accuracy than Arabic light stemmer as it appear in our experiment.

4.2.4 Term Pruning

Pruning, in machine learning, refers to an action of removing non relevant features from the feature space. In text mining, pruning is a useful preprocessing concept because most words in the text corpus are low- frequency words. According to the Zipf's law, given some corpus of natural language texts, if we rank the words according to their frequencies, the distribution of word frequencies is an inverse power law with the exponent of roughly one [83]. This implies that, in any training corpus, the majoritie of the words in the corpus appear only a few times. A word that appears only a few times is usually statistically insignificant low document frequency, low information gain, etc. Moreover, the probability of seeing word, that occurs only once or twice in the training data, in the future document is very low.

In classification, the pruning often yields the smaller size of the feature space, a smaller classification model and a better performance on testing dataset, because of the irrelevant of low frequency words to the text categorization task [84]. The minimum frequency of a word to be included in the word list is varied in each training corpus, and defined by the domain expert.

There are three different methods for pruning [85]:

1. **Perceptual:**

Ignore words that appear in less than below percent percentage of all documents and more than above percent percentage of all documents.

2. **Ranking:**

Words are ordered by frequency and words with a frequency equal or less than the frequency of the rank given by below rank will be pruned, and words with a frequency equal or higher than the frequency of the rank given by above rank will be pruned.

3. **Absolute:**

Ignore words that appear in less than below absolute many documents, and more than that many documents.

We will use perceptual method as it gives the most accurate result as appear in our experiment.

4.2.5 Vector Space Model (VSM) and Term Weighting Schemes

The standard vector space model (VSM) for information retrieval uses vectors to represent documents and the elements of a vector consist of words appearing in the collection. The mathematical representation is given as follows [86]:

$$V_{nm} = \left\{ \begin{matrix} v_{11} & & \cdots & v_{1m} \\ v_{21} & & \cdots & v_{2m} \\ \cdots & \cdots & v_{ij} & \cdots \\ v_{n1} & & \cdots & v_{nm} \end{matrix} \right\} \quad (4.1)$$

The rows of the matrix are defined as documents in the vector space while the columns of the matrix are defined as the terms which are used to describe or index the documents in the vector space. This matrix is commonly referred to as the document-term matrix. An element vij ($1 \leq i \leq n$, $1 \leq j \leq m$) in the document-term matrix reflects the normalized weight of the indexing term tj assigned to the document di. Here n and m are the number of documents and indexing terms in the vector space respectively.

Popular term weighting schemes are [85]:

1. **Binary Term Occurrences (BTO):** This indicates absence or presence of a word with Booleans 0 or 1 respectively.
2. **Term Frequency (TF):** It measures how frequently a term t occurs in a document d. Since every document is different in length, it is possible that a term would appear

much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization:

$$\mathrm{TF} = \frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}} \qquad (4.2)$$

3. **Term Occurrences (TO):** it is the number of occurrences of term t in the document d.
4. **Term Frequency-Inverse Document Frequency (TF-IDF):** the TF-IDF is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. Term frequency tf(t, d) is calculated as in equation 4.2. Document frequency df(t) is number of documents in which the term t occurs at least once [36,37, 52, 55, 59]. The inverse document frequency can be calculated from document frequency using the formula:

$$\mathrm{TFIDF} = \log(\text{num of Docs} / \text{num of docs with word i}) \qquad (4.3)$$

A reasonable measure of term importance may then be obtained by using the product of the term frequency and the inverse document frequency.

$$\mathrm{TFIDF} = tf * idf \qquad (4.4)$$

We will use (TO) because we find it gives the most accurate result.

4.3 Text preprocessing tools

We use RapidMiner (formerly YALE (Yet Another Learning Environment)) for text preprocessing and classification [87]. 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. It is used for business and industrial applications as well as for research, education, training, rapid prototyping, and application development and supports all steps of the data mining process including results visualization, validation and optimization.

RapidMiner provides more than 1,000 operators for all main machine learning procedures, including input and output, and data preprocessing and visualization. Process Documents from files is a RapidMiner operator that Generates word vectors from a text collection stored in multiple files. It also provides different term weighting schemes, and term pruning options [87].

RapidMiner provides a large collection of machine learning algorithms for data preprocessing, classification, clustering, association rules, and visualization, which can be invoked through a common Graphical User Interface. Using RapidMiner we applied preprocessing on both corpus BHA and OCA as follows:

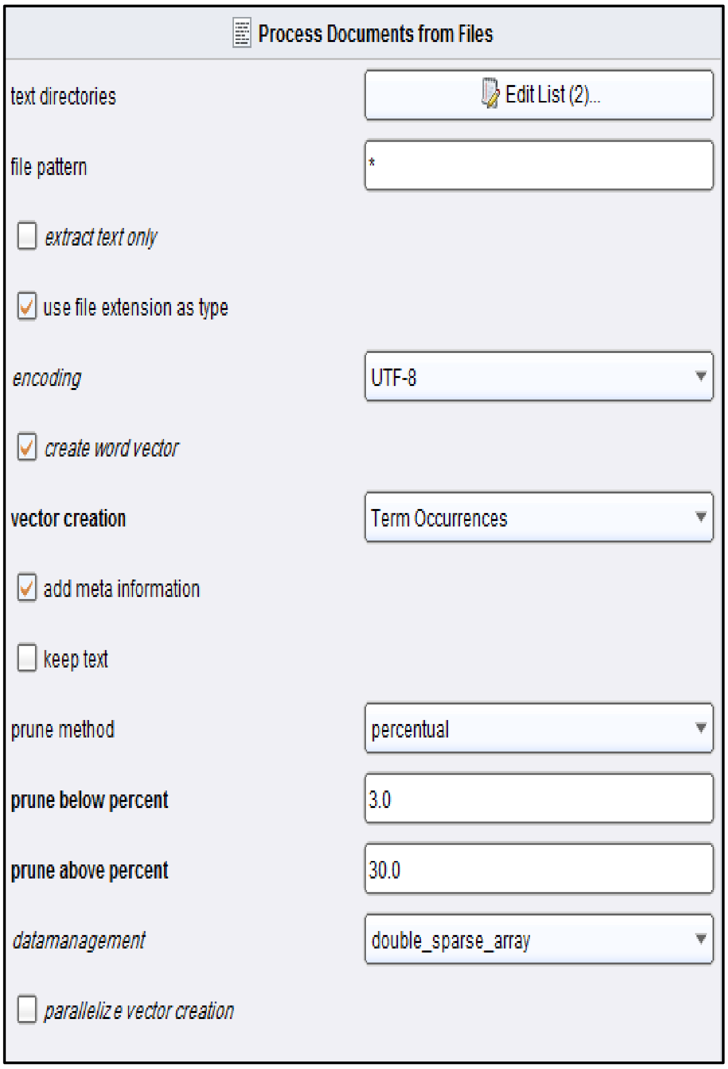
4.4 BHA Preprocessing

This stage is the most important stage, as we try to generate text representations for BHA corpus so that we have made all possible tests in order to obtain higher accuracy by using decision tree classifier. The following table 4.4 illustrates some of these experiments.

Table 4.4: some of the generated text representations for BHA

Term pruning	Vector Creation	Morphological Analysis	accuracy
Perceptual	**TO**	**Arabic Stem**	**90.70**
Perceptual	TF	Arabic Stem Light	78.16
Perceptual	BTO	Arabic Stem	57.14
Perceptual	TF-IDF	Arabic Stem Light	70.72
Ranking	TO	Arabic Stem	62.24
Ranking	TF	Arabic Stem Light	69.21
Ranking	BTO	Arabic Stem	62.70
Ranking	TF-IDF	Arabic Stem Light	58.32
Absolute	TO	Arabic Stem	56.84
Absolute	TF	Arabic Stem Light	69.66
Absolute	BTO	Arabic Stem	69.21
Absolute	TF-IDF	Arabic Stem Light	59.32

In the preprocessing step, the main required Arabic documents process is Process Documents from Files which generates word vectors from a text collection stored in multiple files(BHA) one directory contains positive and another one contains negative, for clearly reading Arabic text files we used UTF-8,for Vector creation we used TO term Weighting Schemes and for term pruning we used perceptual method with minimum threshold 3% and maximum 30%. Figure 4.4 shows a screenshot of Process Documents from Files parameters.



Process Documents from Files

text directories: Edit List (2)...

file pattern: *

□ *extract text only*

☑ use file extension as type

encoding: UTF-8

☑ *create word vector*

vector creation: Term Occurrences

☑ add meta information

□ keep text

prune method: percentual

prune below percent: 3.0

prune above percent: 30.0

datamanagement: double_sparse_array

□ *parallelize vector creation*

Figure 4. 4: Process Documents from Files parameters

We performed five operators; to generate text representations for BHA Corpuse, as in table 4.4 these five operator achieve the higher accuracy 90.70, These operator are Tokenize we specified none letters mode for splitting points as described previously in section 4.2.1, filters Arabic stopwords as described previously in section 4.2.2, Stem Arabic as described previously in section 4.2.3, Filter tokens (by length) operator that filters tokens based on their length (i.e. the number of characters they contain) (min characters = 2, max characters = 25)., and finally Filter documents (by content) operator: - Filters documents from a document collection based on their contents. A document is kept in the collection, if it does not contain match [a-zA-Z]. Figure 4.5 shows a screenshot of the five operators, Figure 4.6 shows the process of transforming BHA text documents to record using RapidMiner, from this figure we can see that the number of examples is decreased from 8224 to 6934 and the number of attributes is 228 these are occurred due to preprocessing stages in figure 4.5. Figure 4.7 shows the resulting wordlist (dictionary).

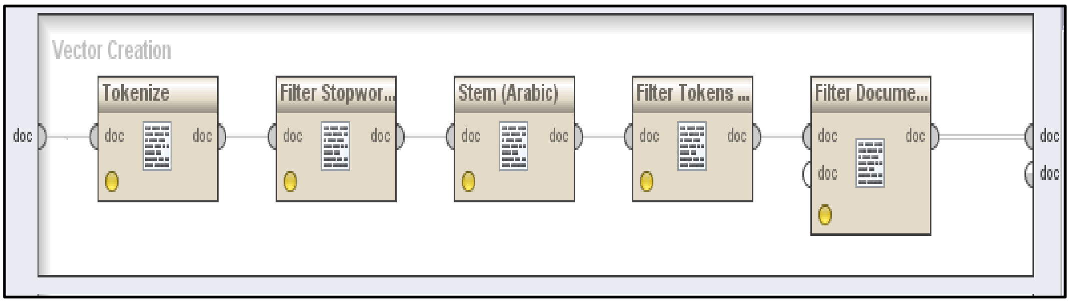
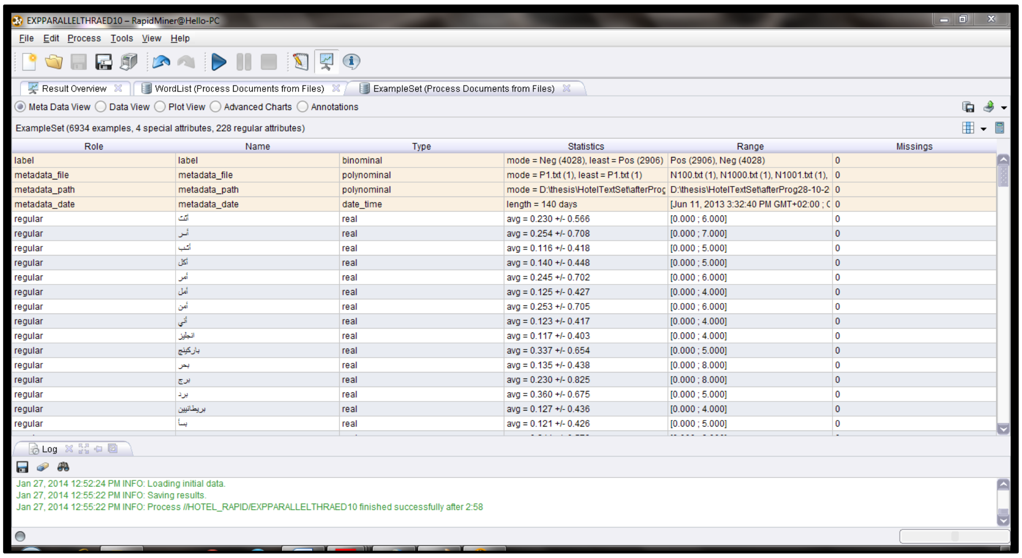


Figure 4. 5: BHA vector creation preprocessing


EXPPARALLELTHRAED10 – RapidMiner@Hello-PC
File Edit Process Tools View Help
Result Overview
WordList (Process Documents from Files)
ExampleSet (Process Documents from Files)
Log

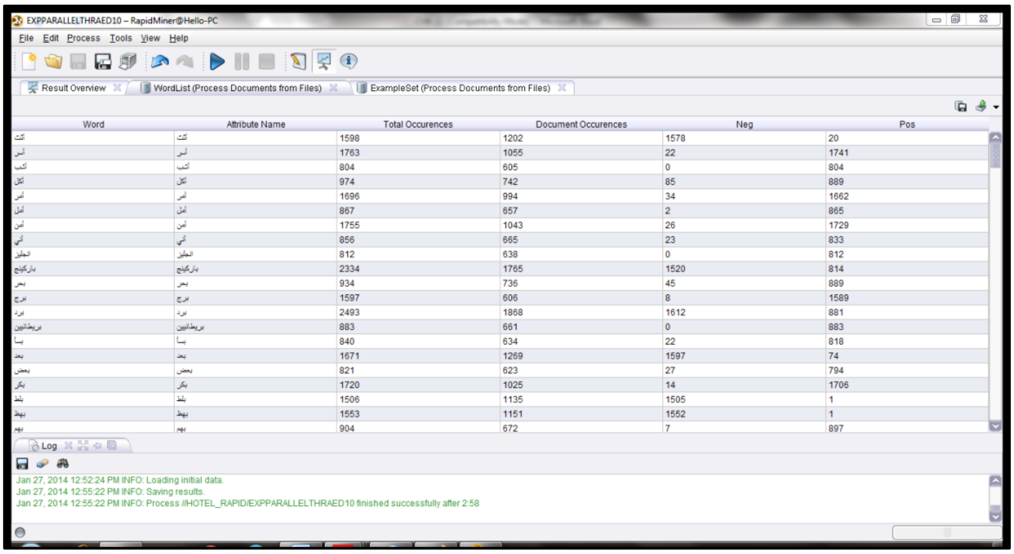
● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (6934 examples, 4 special attributes, 228 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	label	binominal	mode = Neg (4028), least = Pos (2906)	Pos (2906), Neg (4028)	0
metadata_file	metadata_file	polynominal	mode = P1.txt (1), least = P1.txt (1)	N100.txt (1), N1000.txt (1), N1001.txt (1),	0
metadata_path	metadata_path	polynominal	mode = D:\thesis\HotelTextSet\afterProg	D:\thesis\HotelTextSet\afterProg28-10-2	0
metadata_date	metadata_date	date_time	length = 140 days	[Jun 11, 2013 3:32:40 PM GMT+02:00 ; (	0
regular	أثث	real	avg = 0.230 +/- 0.566	[0.000 ; 6.000]	0
regular	أسر	real	avg = 0.254 +/- 0.708	[0.000 ; 7.000]	0
regular	أشب	real	avg = 0.116 +/- 0.418	[0.000 ; 5.000]	0
regular	أكل	real	avg = 0.140 +/- 0.448	[0.000 ; 5.000]	0
regular	أمر	real	avg = 0.245 +/- 0.702	[0.000 ; 6.000]	0
regular	أمل	real	avg = 0.125 +/- 0.427	[0.000 ; 4.000]	0
regular	أمن	real	avg = 0.253 +/- 0.705	[0.000 ; 6.000]	0
regular	أني	real	avg = 0.123 +/- 0.417	[0.000 ; 4.000]	0
regular	انجليز	real	avg = 0.117 +/- 0.403	[0.000 ; 4.000]	0
regular	باركينج	real	avg = 0.337 +/- 0.654	[0.000 ; 5.000]	0
regular	بحر	real	avg = 0.135 +/- 0.438	[0.000 ; 8.000]	0
regular	برج	real	avg = 0.230 +/- 0.825	[0.000 ; 8.000]	0
regular	برد	real	avg = 0.360 +/- 0.675	[0.000 ; 5.000]	0
regular	بريطانيين	real	avg = 0.127 +/- 0.436	[0.000 ; 4.000]	0
regular	بسأ	real	avg = 0.121 +/- 0.426	[0.000 ; 5.000]	0

```
Jan 27, 2014 12:52:24 PM INFO: Loading initial data.
Jan 27, 2014 12:55:22 PM INFO: Saving results.
Jan 27, 2014 12:55:22 PM INFO: Process //HOTEL_RAPID/EXPPARALLELTHRAED10 finished successfully after 2:58
```

Figure 4. 6: BHA Transforming text documents to Example Set using RapidMiner


EXPPARALLELTHRAED10 – RapidMiner@Hello-PC
File Edit Process Tools View Help
Result Overview
WordList (Process Documents from Files)
ExampleSet (Process Documents from Files)
Log

Word	Attribute Name	Total Occurences	Document Occurences	Neg	Pos
أثث	أثث	1598	1202	1578	20
أسر	أسر	1763	1055	22	1741
أشب	أشب	804	605	0	804
أكل	أكل	974	742	85	889
أمر	أمر	1696	994	34	1662
أمل	أمل	867	657	2	865
أمن	أمن	1755	1043	26	1729
أني	أني	856	665	23	833
انجليز	انجليز	812	638	0	812
باركينج	باركينج	2334	1765	1520	814
بحر	بحر	934	736	45	889
برج	برج	1597	606	8	1589
برد	برد	2493	1868	1612	881
بريطانيين	بريطانيين	883	661	0	883
بسأ	بسأ	840	634	22	818
بعد	بعد	1671	1269	1597	74
بعض	بعض	821	623	27	794
بكر	بكر	1720	1025	14	1706
بلط	بلط	1506	1135	1505	1
بهط	بهط	1553	1151	1552	1
بهم	بهم	904	672	7	897

```
Jan 27, 2014 12:52:24 PM INFO: Loading initial data.
Jan 27, 2014 12:55:22 PM INFO: Saving results.
Jan 27, 2014 12:55:22 PM INFO: Process //HOTEL_RAPID/EXPPARALLELTHRAED10 finished successfully after 2:58
```

Figure 4. 7: BHA Transforming text documents to word list using RapidMiner

4.5 OCA Preprocessing

This stage is the most important stage, as we try to generate text representations for OCA corpus so that we have made all possible tests in order to obtain higher accuracy by using decision tree classifier. The following table 4.5 illustrates some of these experiments.

Table 4.5: some of the generated text representations for OCA

Term pruning	Vector Creation	Morphological Analysis	accuracy
Perceptual	TO	Arabic Stem	50.12
Perceptual	TF	Arabic Stem Light	55.62
Perceptual	BTO	Arabic Stem	69.14
Perceptual	TF-IDF	Arabic Stem Light	78.50
Ranking	TO	Arabic Stem	58.63
Ranking	TF	Arabic Stem Light	68.28
Ranking	BTO	Arabic Stem	87.50
Ranking	TF-IDF	Arabic Stem Light	81.48
Absolute	TO	Arabic Stem	75.00
Absolute	TF	Arabic Stem Light	74.07
Absolute	BTO	Arabic Stem	88.89
Absolute	TF-IDF	Arabic Stem Light	87.50
Perceptual	**TO**	**without**	**92.59**
Perceptual	TF	without	87.88
Perceptual	BTO	without	89.09
Perceptual	TF-IDF	without	87.21

To generate word vectors from a text collection stored in multiple files (OCA), one directory contains positive and another one contains negative, For clearly reading Arabic text files we used UTF-8, for Vector creation we used TO term Weighting Schemes and for term pruning we used perceptual method with minimum threshold 3% and maximum 30%. Figure 4.4 shows a screenshot of Process Documents from Files parameters.

We performed three operators; to generate text representations for OCA Corpuse. These operator are Tokenize. We spesified none letters mode for splitting points as described previously in section 4.2.1, filters Arabic stopwords as described previously in section 4.2.2, and lastly Filter documents (by content) operator: - Filters documents from a document collection based on their contents. A document is kept in the collection, if it does not contain match [a-zA-Z].Figure 4.8 shows a screenshot of the three operators, Figure 4.9 shows the process of transforming OCA text documents to record using RapidMiner, from this figure we can see that the number of examples is decreased from 500 to 135 and the number of attributes is 1118 these are occurred due to preprocessing stages in figure 4.8. Figure 4.10 shows the resulting wordlist (dictionary).

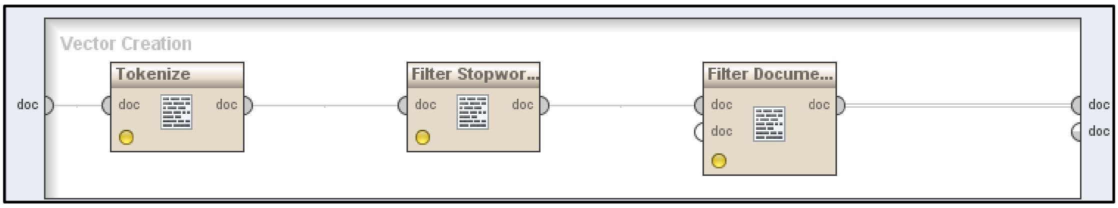
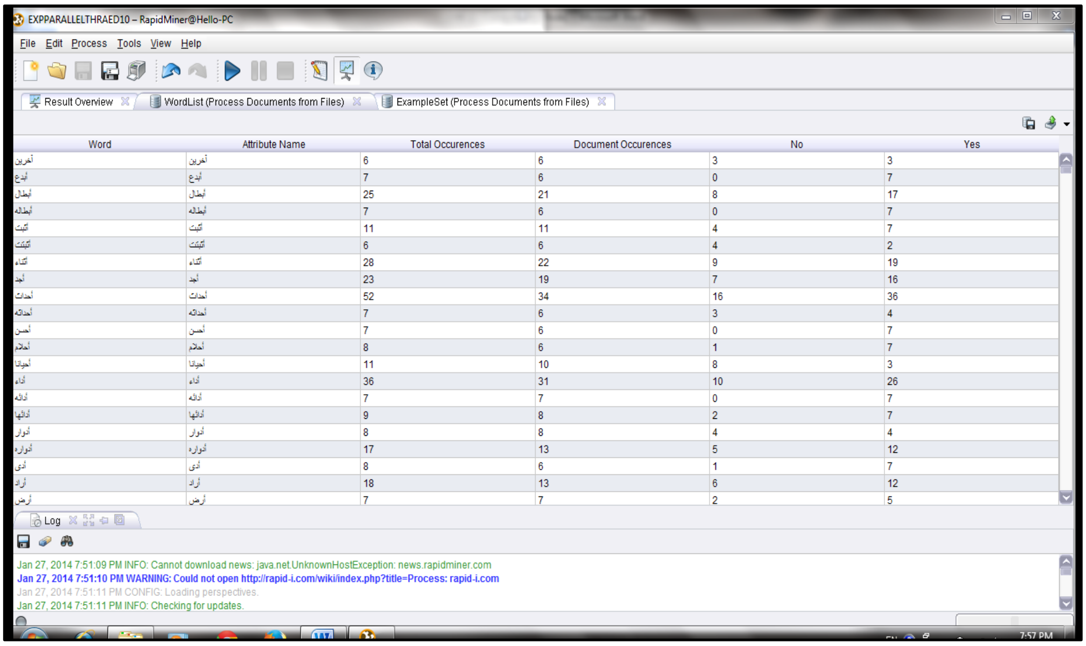


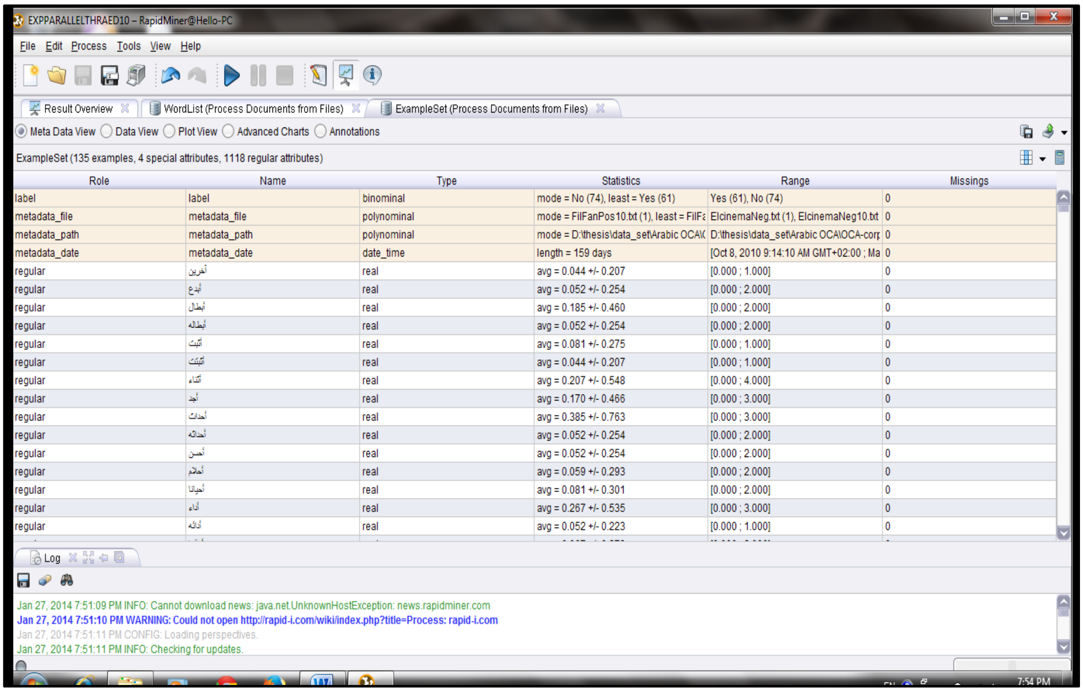
Figure 4. 8: OCA vector creation preprocessing.


EXPPARALLELTHRAED10 – RapidMiner@Hello-PC
File Edit Process Tools View Help
Result Overview
WordList (Process Documents from Files)
ExampleSet (Process Documents from Files)
Log

Word	Attribute Name	Total Occurences	Document Occurences	No	Yes
آخرين	آخرين	6	6	3	3
أبدع	أبدع	7	6	0	7
أبطال	أبطال	25	21	8	17
أبطاله	أبطاله	7	6	0	7
أثبت	أثبت	11	11	4	7
أثبتت	أثبتت	6	6	4	2
أثناء	أثناء	28	22	9	19
أجد	أجد	23	19	7	16
أحداث	أحداث	52	34	16	36
أحداثه	أحداثه	7	6	3	4
أحسن	أحسن	7	6	0	7
أحلام	أحلام	8	6	1	7
أحيانا	أحيانا	11	10	8	3
أداء	أداء	36	31	10	26
أدائه	أدائه	7	7	0	7
أدائها	أدائها	9	8	2	7
أدوار	أدوار	8	8	4	4
أدواره	أدواره	17	13	5	12
أدى	أدى	8	6	1	7
أراد	أراد	18	13	6	12
أرض	أرض	7	7	2	5

```
Jan 27, 2014 7:51:09 PM INFO: Cannot download news: java.net.UnknownHostException: news.rapidminer.com
Jan 27, 2014 7:51:10 PM WARNING: Could not open http://rapid-i.com/wiki/index.php?title=Process: rapid-i.com
Jan 27, 2014 7:51:11 PM CONFIG: Loading perspectives.
Jan 27, 2014 7:51:11 PM INFO: Checking for updates.
```

Figure 4. 9: OCA Transforming text documents to word list using RapidMiner


EXPPARALLELTHRAED10 – RapidMiner@Hello-PC
File Edit Process Tools View Help
Result Overview
WordList (Process Documents from Files)
ExampleSet (Process Documents from Files)
Log

● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (135 examples, 4 special attributes, 1118 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	label	binominal	mode = No (74), least = Yes (61)	Yes (61), No (74)	0
metadata_file	metadata_file	polynominal	mode = FilFanPos10.txt (1), least = FilFa	ElcinemaNeg.txt (1), ElcinemaNeg10.txt	0
metadata_path	metadata_path	polynominal	mode = D:\thesis\data_set\Arabic OCA\(	D:\thesis\data_set\Arabic OCA\OCA-corp	0
metadata_date	metadata_date	date_time	length = 159 days	[Oct 8, 2010 9:14:10 AM GMT+02:00 ; Ma	0
regular	آخرين	real	avg = 0.044 +/- 0.207	[0.000 ; 1.000]	0
regular	أبدع	real	avg = 0.052 +/- 0.254	[0.000 ; 2.000]	0
regular	أبطال	real	avg = 0.185 +/- 0.460	[0.000 ; 2.000]	0
regular	أبطاله	real	avg = 0.052 +/- 0.254	[0.000 ; 2.000]	0
regular	أثبت	real	avg = 0.081 +/- 0.275	[0.000 ; 1.000]	0
regular	أثبتت	real	avg = 0.044 +/- 0.207	[0.000 ; 1.000]	0
regular	أثناء	real	avg = 0.207 +/- 0.548	[0.000 ; 4.000]	0
regular	أجد	real	avg = 0.170 +/- 0.466	[0.000 ; 3.000]	0
regular	أحداث	real	avg = 0.385 +/- 0.763	[0.000 ; 3.000]	0
regular	أحداثه	real	avg = 0.052 +/- 0.254	[0.000 ; 2.000]	0
regular	أحسن	real	avg = 0.052 +/- 0.254	[0.000 ; 2.000]	0
regular	أحلام	real	avg = 0.059 +/- 0.293	[0.000 ; 2.000]	0
regular	أحيانا	real	avg = 0.081 +/- 0.301	[0.000 ; 2.000]	0
regular	أداء	real	avg = 0.267 +/- 0.535	[0.000 ; 3.000]	0
regular	أدائه	real	avg = 0.052 +/- 0.223	[0.000 ; 1.000]	0

```
Jan 27, 2014 7:51:09 PM INFO: Cannot download news: java.net.UnknownHostException: news.rapidminer.com
Jan 27, 2014 7:51:10 PM WARNING: Could not open http://rapid-i.com/wiki/index.php?title=Process: rapid-i.com
Jan 27, 2014 7:51:11 PM CONFIG: Loading perspectives.
Jan 27, 2014 7:51:11 PM INFO: Checking for updates.
```

Figure 4. 10: OCA Transforming text documents to Example Set using RapidMiner

4.6 Summary

In this chapter, we described the Data collection and preprocess stages, we collected two Arabic corpus and made an overall description about them, these collections to be classified need to be processed so that we apply preprocessing stages including feature reduction using morphological analysis techniques, and term weighting. In the next chapter we will go to the next stages which is experimental result and evaluations.

Chapter 5

Experimental Results and Evaluation

Opinion mining may be seen as classification problem where review is classified as positive or negative. In this chapter, we describe applying a family of decision trees learning algorithms such as decision tree, Quinlan's ID3 numerical, and decision stumps on our data sets. In addition, we present and analyze the experimented results after learning OCA and BHA corpus by the sequential and parallel decision trees learning algorithms. Also, we make a comparison with other classification methods such as KNN, NB, SVM .

5.1 Opinion Mining Classification Experiments

Classification "is a data mining and machine learning technique used to predict group membership for data instances" [88]. The goal of classification is to predict accurately the target class for each case in the data [89]. To achieve this goal we perceptually split OCA and BHA into two disjoint sets 80% training set and 20% test set, training set used to build the model and test set used to validate it. There are many types of classification algorithms; we go to use decision trees to get more understandable rules and efficient results about why results appear so. For booth corpuses, BHA and OCA, we applied the following decision trees classification algorithms decision tree, Quinlan's ID3 numerical, and decision stumps.

5.2 BHA Data Mining Classification Experiments

To get understandable rule from BHA corpus which is split into two parts; 80% of the corpus for training and the remaining 20% for testing using stratified sampling which keeps class distributions and remains the same after splitting. We learned it using the following three sequential decision trees classification algorithms:

5.2.1 Decision Tree classifier

It ran with the following parameters that gives the most accurate result:

- **Decision Tree Criterion**: gain ratio criteria in order to select attributes and numerical splits.
- **Minimal size for split**: a minimal size of four for a node to allow a split.
- **Minimal leaf size**: a minimal size of two for all leaves.
- **Minimal gain**: a minimal gain of 0.1 to produce a split to pick a good attribute for the root of the tree and give us tree with the greatest predictive accuracy.
- and finally with disabled the pre-pruning and pruning to deliver a tree without any pre-pruning nor pruning because the pruning reduces the size of decision tree Which leads to the loss of important information derived from tree rules also the accuracy decreases from 90.7% to 86.37% as shown in figure 5.1 and figure 5.5

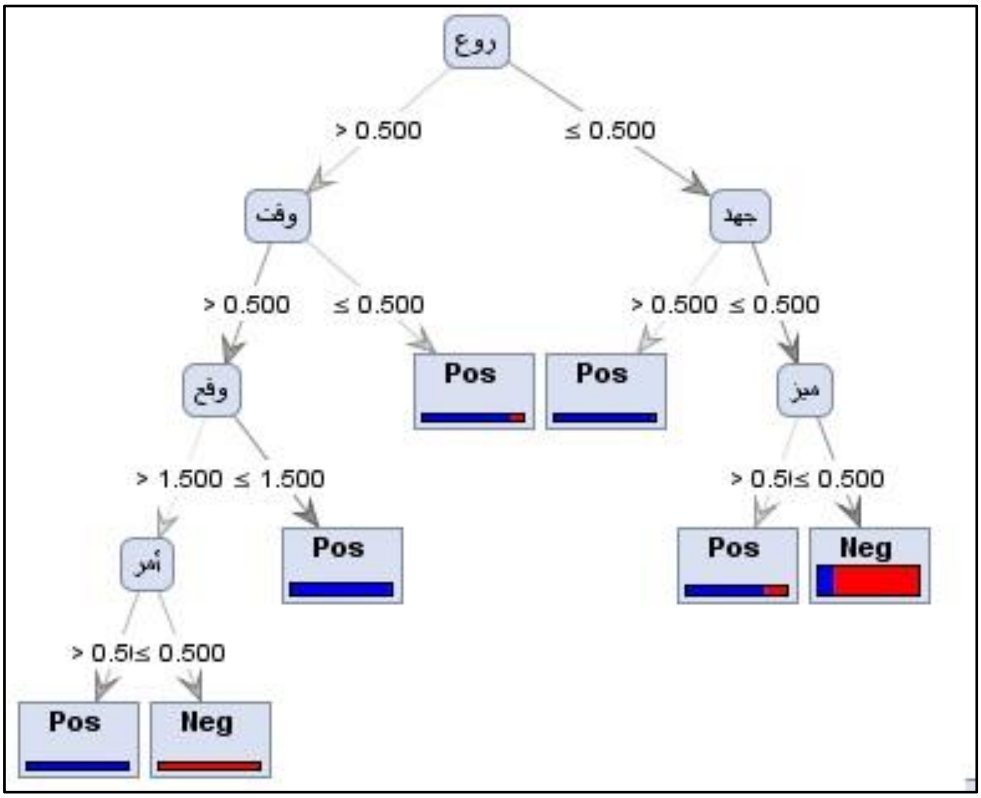


Figure 5.1: the BHA constructed tree with pruning

In this figure we enabled pruning so that we get 86.37% accuracy, we can see that the derived rules are not clearly as it is in less details.

5.2.2 Quinlan's ID3 numerical: -

This classifier run with the gain ratio criteria, a minimal size of 4 for a node to allow a split, a minimal size of 2 for all leaves, and a minimal gain of 0.1. This classification method is shown in figure 5.2

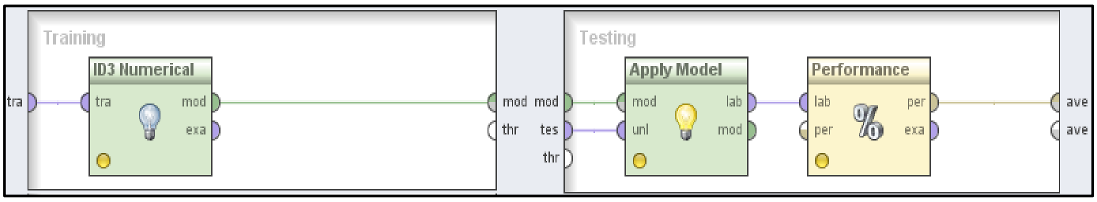


Figure 5.2: applying ID3 classifier on BHA

5.2.3 Decision stumps:

This classifier run with the gain ratio criteria and a minimal size of 1 for all leaves.

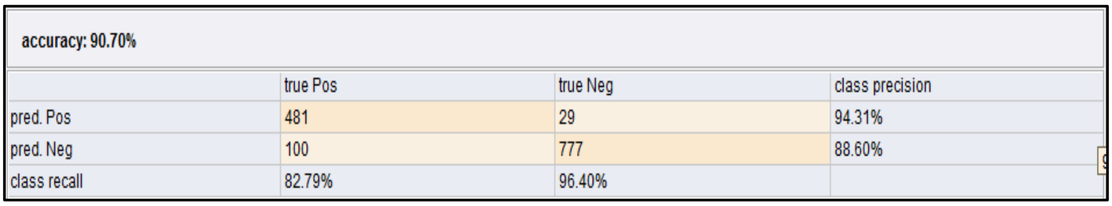
5.2.4 BHA Results:

To perform the comparisons of the tested algorithms, the accuracy assessment reflects really the difference between our classification and the reference data. The performance of each classifier was evaluated by using the accuracy and F-measure, which is stated in table 5.1, each experiment is made 5 times and then we calculated the mean of them.

Table 5.1: BHA Data Mining sequential process

Algorithm	Time (sec)	Accuracy	Mean Recall	Mean Precision	F-Measure
Decision Tree	417.00	90.70	90.94	90.94	90.94
Decision Stump	387.00	74.91	71.08	85.65	77.69
ID3 Numerical	3378.00	90.05	89.78	89.78	89.78

We note from table 5.1 that Decision tree algorithm has the maximum accuracy of (90.70%) and F-Measure (90.94%). The worst results was Decision Stump with accuracy of (74.91%) and f-measure (77.69) .The results of accuracy assessment are summarized in a confusion matrix as shown in figure 5.3 that show the predictions about 20% of 6934 examples which is 1386 examples.



accuracy: 90.70%

	true Pos	true Neg	class precision
pred. Pos	481	29	94.31%
pred. Neg	100	777	88.60%
class recall	82.79%	96.40%	

Figure 5.3: BHA Performance Vector

For, the time performance the best is Decision Stump with 387.00 sec. and worst is 3378 sec with ID3. From this, we conclude that the best classifier is Decision Tree, which has the best accuracy, f-measure and acceptable time (417sec).

From the above decision tree experiment we constructed the following BHA Tree: as in figure 5.4.

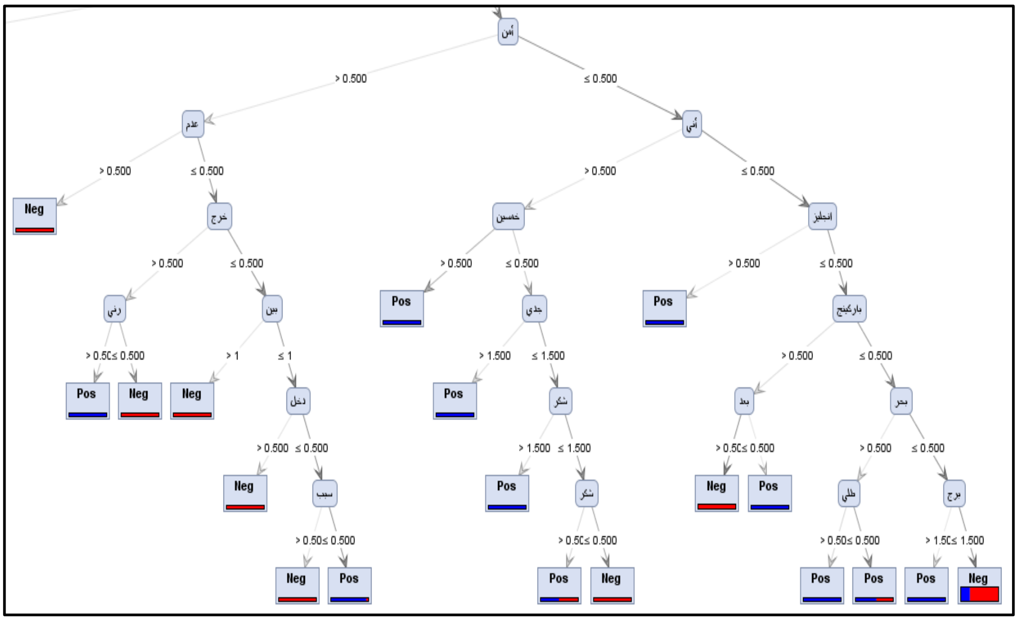


Figure 5.4: BHA constructed Tree without pruning

In this figure we constructed the tree without any pre-pruning nor pruning which lead to have important information as we got the rules in more details.

In addition, we extracted the BHA Tree Rules: as in figure 5.5

```
وقت> 0.500
|   غرف> 0.500
|   |   جمل> 0.500
|   |   |   أمن> 0.500: Neg {Pos=0, Neg=1}
|   |   |   أمن≤ 0.500
|   |   |   |   بسأ> 0.500: Neg {Pos=0, Neg=1}
|   |   |   |   بسأ≤ 0.500
|   |   |   |   |   بكر> 0.500: Neg {Pos=0, Neg=1}
|   |   |   |   |   بكر≤ 0.500: Pos {Pos=10, Neg=0}
|   |   جمل≤ 0.500
|   |   |   مرر> 0.500: Neg {Pos=0, Neg=11}
|   |   |   مرر≤ 0.500
|   |   |   |   سوا> 0.500: Neg {Pos=0, Neg=5}
|   |   |   |   سوا≤ 0.500
|   |   |   |   |   نسب> 0.500: Pos {Pos=5, Neg=0}
|   |   |   |   |   نسب≤ 0.500
|   |   |   |   |   |   ميز> 0.500
|   |   |   |   |   |   |   جود> 0.500: Neg {Pos=0, Neg=2}
|   |   |   |   |   |   |   جود≤ 0.500: Pos {Pos=5, Neg=0}
|   |   |   |   |   |   ميز≤ 0.500
|   |   |   |   |   |   |   بعد> 1.500: Pos {Pos=1, Neg=0}
|   |   |   |   |   |   |   بعد≤ 1.500
|   |   |   |   |   |   |   |   جود> 0.500: Pos {Pos=1, Neg=0}
|   |   |   |   |   |   |   |   جود≤ 0.500: Neg {Pos=0, Neg=11}
```

Figure 5.5: BHA Tree Rules

The rules in figure 5.5 are extracted from figure 5.4.

5.3 OCA Data Mining Classification Experiments

Also to extract understandable rule from OCA corpus which is split into two parts; 80% of the corpus for training and the remaining 20% for testing using shuffled sampling which distributes data randomly. We learned it using the following sequential decision trees classification algorithms:

5.3.1 Decision Tree:

This classifier executed with the gain ratio, a minimal size of 4 for node splitting, a minimal size of 2 for all leaves, a minimal gain of 0.1, and lastly disabling the pre-pruning and pruning to deliver a tree without any pre-pruning nor pruning. This done because the pruning minimizes the tree which leads to the loss of important information derived from tree rules as shown in figure 5.6, in both cases disable and enable pruning the accuracy become is the same 92.59%.

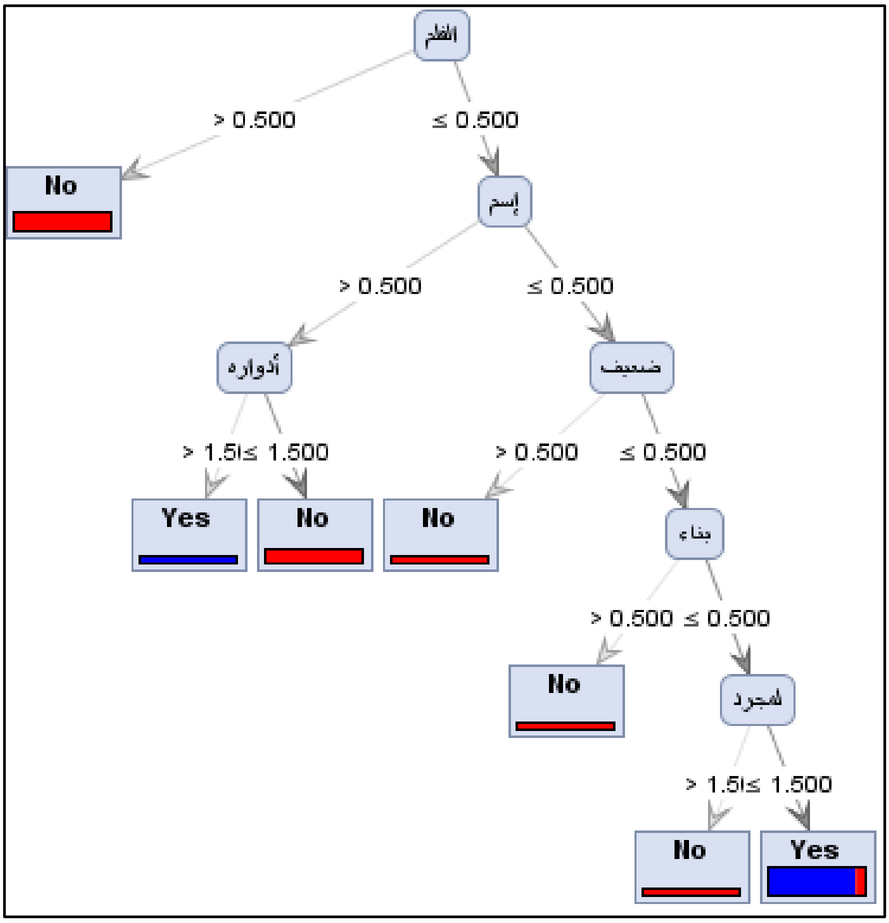


Figure 5.6: the OCA constructed tree with pruning

In this figure we enabled pruning so that we get 92.59% accuracy, we can see that the derived rules are not clearly as it is in less details.

5.3.2 Quinlan's ID3 numerical:

This classifier executed with the gain ratio criteria, a minimal size of four for a node to allow a split, a minimal size of two for all leaves, and a minimal gain of 0.1.

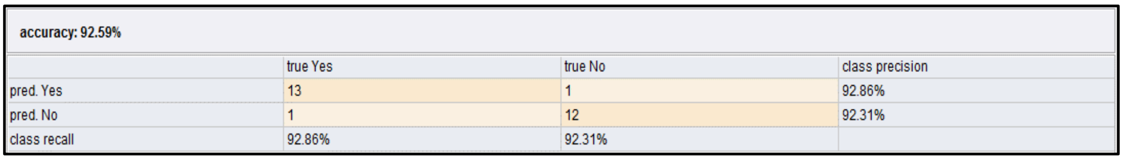
5.3.3 Decision stumps:

this classifier is executed with the gain ratio criteria and a minimal size of 1for all leaves.

Table 5.2: OCA Data Mining sequential process

Algorithm	Time (sec)	Accuracy	Mean Recall	Mean Precision	F-Measure
Decision Tree	68.00	92.59	92.58	92.58	92.58
Decision Stump	8.00	77.78	76.92	85.00	80.76
ID3 Numerical	65.00	92.59	92.58	92.58	92.58

From table 5.2 we noted that Decision tree algorithm and ID3 have the maximum accuracy of (92.59%) and F-Measure (92.58%) The worst results was Decision Stump with accuracy of (74.78%) and f-measure (80.76). The results of accuracy assessment are summarized in a confusion matrix as shown in figure 5.7 that show the predictions about 20% of 135 examples which is 27 examples.



accuracy: 92.59%

	true Yes	true No	class precision
pred. Yes	13	1	92.86%
pred. No	1	12	92.31%
class recall	92.86%	92.31%	

Figure 5.7: OCA Performance Vector

When we compare the results based on the time performance, we found that the worst result is Decision tree with 68 sec and the best is decision stump with 8 sec. In addition, by differentiation of the results based on accuracy we found that the best are ID3 and decision tree with 92.59% and the worst case is decision stump with 77.78%

It is clear that this confusion matrix contains information about realistic and predicted OCA classifications. In order to evaluate our experiment we used a common way of evaluating results of Language and Learning experiments using Recall, Precision and F-measure.

The constructed tree after applying decision tree operator are in figure 5.8

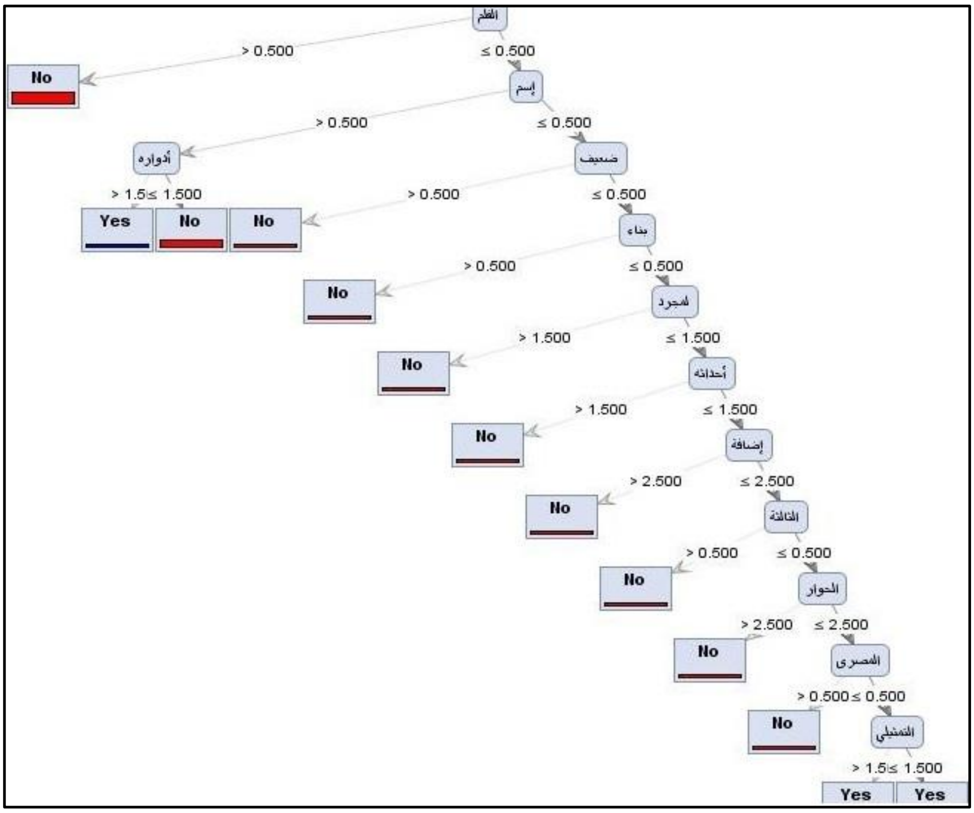


Figure 5.8: OCA Tree

In this figure5.8 we constructed the tree without any pre-pruning nor pruning which lead to have important information as we got the rules in more details.

From the above OCA experiment, we extracted the following OCA Tree Rules as illustrated in figure 5.9:

```
الفلم≤ 0.500
|    إسم> 0.500
|    |    أدواره> 1.500: Yes {Yes=2, No=0}
|    |    أدواره≤ 1.500: No {Yes=0, No=22}
|    إسم≤ 0.500
|    |    ضعيف> 0.500: No {Yes=0, No=4}
|    |    ضعيف≤ 0.500
|    |    |    بناء> 0.500: No {Yes=0, No=3}
|    |    |    بناء≤ 0.500
|    |    |    |    لمجرد> 1.500: No {Yes=0, No=2}
|    |    |    |    لمجرد≤ 1.500
|    |    |    |    |    أحداثه> 1.500: No {Yes=0, No=1}
|    |    |    |    |    أحداثه≤ 1.500
|    |    |    |    |    |    إضافة> 2.500: No {Yes=0, No=1}
|    |    |    |    |    |    إضافة≤ 2.500
|    |    |    |    |    |    |    الثالثة> 0.500: No {Yes=0, No=1}
|    |    |    |    |    |    |    الثالثة≤ 0.500
|    |    |    |    |    |    |    |    الحوار> 2.500: No {Yes=0, No=1}
|    |    |    |    |    |    |    |    الحوار≤ 2.500
|    |    |    |    |    |    |    |    |    المصرى> 0.500: No {Yes=0, No=1}
|    |    |    |    |    |    |    |    |    المصرى≤ 0.500
|    |    |    |    |    |    |    |    |    |    التمثيلي> 1.500: Yes {Yes=1, No=1}
|    |    |    |    |    |    |    |    |    |    التمثيلي≤ 1.500: Yes {Yes=58,No=0}
```

Figure 5.9: OCA Tree Rules

The rules in figure 5.9 is extracted from figure 5.8

5.4 BHA Parallel opinion mining

To improve decision trees time performance , to achieve a higher classification results, and to evaluate the performance of the parallel classifier purposes we have applied three decision trees classification algorithms which are parallel decision tree, parallel Quinlan's ID3 numerical, and parallel decision stumps with the same sequential parameter.

We have executed the parallel classifiers varying the number of threads from 2 to 12, varying the parallel classifiers to observe the effects of different classifier on the performance, and we repeated each experiment five times to get the average accuracy, recall, precision, and f-measure, which recorded in table 5.3.

Table 5.3: BHA Parallel Accuracy and F-Measure

Algorithm	THREAD NO	Accuracy	Recall	Precision	F-Measure
Decision Tree (Parallel)	2	89.83	84.75	90.40	87.48
	3	89.69	88.03	90.42	89.21
	4	**90.63**	88.15	90.46	**89.29**
	6	89.11	88.18	89.07	88.62
	8	87.89	87.05	88.92	87.98
	10	89.11	87.58	90.09	88.82
	12	87:67	85.56	89.67	87.57
Decision Stump (Parallel)	2	74.91	70.10	85.65	77.10
	3	74.91	70.10	85.53	77.05
	4	74.91	70.10	85.65	77.10
	6	79.74	75.96	85.65	80.51
	8	74.91	70.10	85.53	77.05
	10	74.91	70.10	85.53	77.05
	12	74.91	70.10	85.53	77.05
ID3 Numerical (Parallel)	2	90.19	89.53	89.60	89.56
	3	88.75	89.77	89.81	89.79
	4	90.27	89.94	89.95	89.94
	6	89.98	89.35	89.43	89.39
	8	88.90	89.43	89.48	89.45
	10	89.91	89.97	89.97	89.97
	12	**90.70**	89.97	89.99	**89.98**

From table 5.3 we note that in the case of applying parallel decision tree we get the best results of accuracy (90.63%) and f-measure (89.29) at thread 4, these results are different from sequential that have accuracy (90.70%) and f-measure (90.94), In the case of applying parallel decision stump the accuracy (74.91%) and f-measure (77.69) in all threads which is the same as sequential, and in the case of applying parallel ID3 the accuracy (90.70%) and f-measure (89.98) at thread 12 which is greater than sequential accuracy (90.05%) and f-measure (89.78).The results of accuracy are illustrated in figure 5.10.

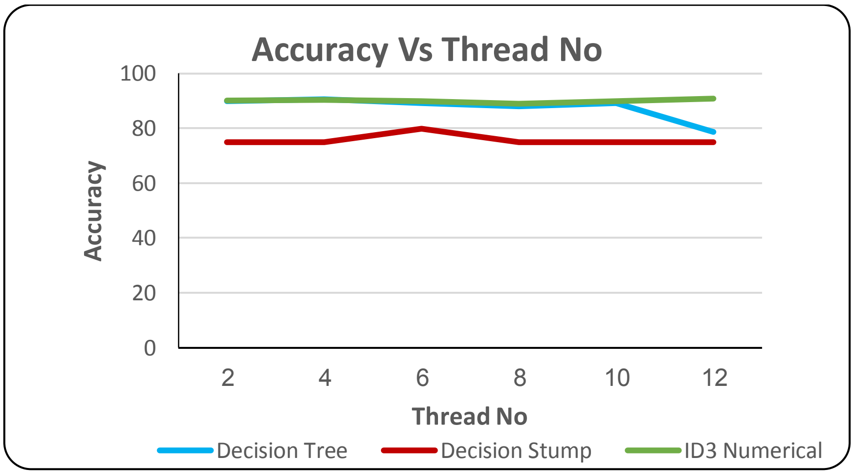


Figure 5.10: The curves of Accuracy for the parallel classifier for BHA

The execution time in seconds recorded in Table 5.4

Table 5.4: BHA Parallel Execution Time

THREAD NO	Parallel Decision Tree Time (sec)	Parallel Decision Stump Time (sec)	Parallel ID3 Time (sec)
2	352.0000	320.0000	3350.000
3	317.0000	262.0000	3300.000
4	**219.0000**	250.0000	3299.000
6	232.0000	**130.0000**	3268.000
8	241.0000	225.0000	2405.000
10	269.0000	233.0000	2330.000
12	318.0000	238.0000	**2300.000**

Several observations can be made by analyzing the results in Table 5.4.**First**, the BHA Decision Tree serial process takes more time than the parallel version. In the parallel version; the execution time decreases when the number of threads increases to reach 4 but it increase when the number of threads increases from 6 to 12. However, the parallel achieves a good execution time compared to a serial one. Figure 5.6 shows the curves of execution time for the parallel Decision Tree classifier on the BHA corpus. The time curve decreases from 1 thread until using 4 threads.We note from figure 5.11 that the serial Decision Tree classification algorithm spent a lot of time classifying the text documents, and the parallel version of it clearly reduces the serial time. Notice that the serial Decision Tree classification algorithm takes about 7 minutes to classify this collection, while the parallel classifier reduces this time to 3 minutes on 4 threads.

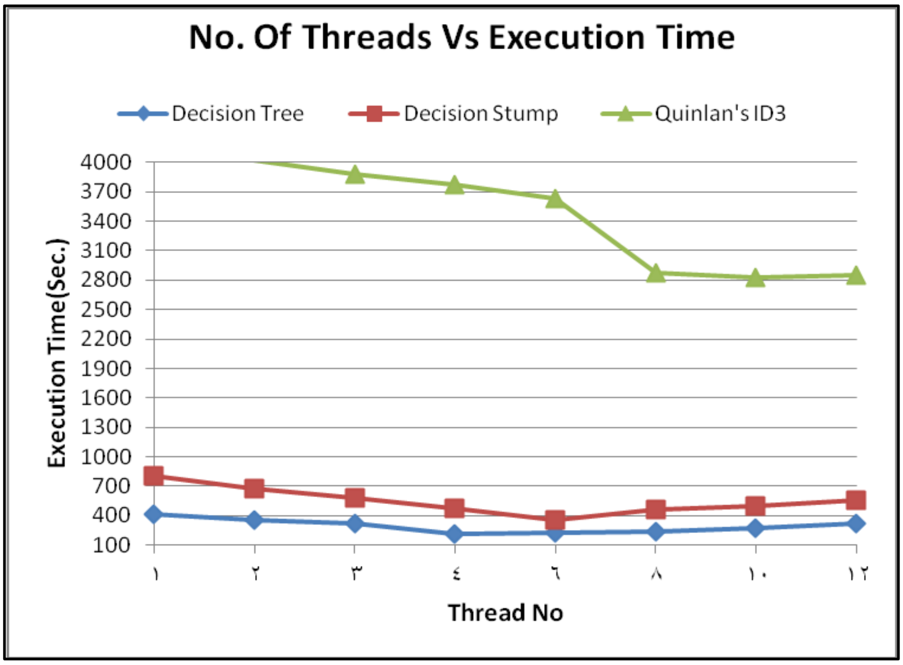


Figure 5.11: The curves of execution time for the parallel classifier for BHA

Second, the BHA Decision Stump serial process takes less time than the parallel version. In the parallel version; the execution time decreases when the number of threads increases to reach 6 but increases when the number of threads increases from 8 to 12. However, the parallel achieves a good execution time compared to serial one. Figure 5.11 shows the curves of execution time for the parallel Decision Stump classifier on the BHA corpus. The time curve decreases from 1 thread until using 6 threads.

We note from Figure 5.11 that the serial Decision Stump classification algorithm spent a lot of time classifying the text documents, and the parallel version of it clearly reduces the serial time. Notice that the serial Decision Stump classification algorithm takes about 6 minutes to classify this collection, while the parallel classifier reduces this time to 2 minutes on 6 threads.

Third, the BHA Decision Quinlan's ID3 numerical serial process takes less time than the parallel version. In the parallel version; the execution time decreases when the number of threads increases. However, the parallel achieves a good execution time compared to serial one. Figure 5.11 shows the curves of execution time for the parallel Decision Quinlan's ID3 numerical classifier on the BHA corpus. The time curve decreases from 1 thread until using 6 threads.

We note from Figure 5.11 that the serial Decision Quinlan's ID3 numerical classification algorithm spent a lot of time classifying the text documents, and the parallel version of it clearly reduces the serial time. Notice that the serial Decision Quinlan's ID3 numerical classification algorithm takes about 7 minutes to classify this collection, while the parallel classifier reduces this time to 2 minutes on 6 threads.

Also, we compute the speedup which refers to how much a parallel algorithm is faster than a corresponding sequential algorithm. The speedup in seconds recorded in Table 5.5. Figure 5.12 demonstrates the relative speedup

Table 5.5: BHA Relative Speedup of the Parallel Classifiers

Parallel Algorithm Thread No	Decision Tree	Decision Stump	ID3 Numerical
2	1.185	1.209	1.008
3	1.315	1.477	1.024
4	**1.904**	1.548	1.024
6	1.797	**2.977**	1.034
8	1.730	1.720	1.405
10	1.550	1.661	1.450
12	1.311	1.626	**1.469**

By analyzing the resulted speedup from table (5.5), we note that when we applied Decision Tree algorithm we get the best speed up 1.904 when used 4 threads, and the best speed up for applying decision stump is 2.977 when used 6 threads, and the best speed up for applying ID3 numerical is 1.469 when used 12 threads.

From the above notes, we found that the Decision Tree classifier achieved the best speedup at 4 threads.

The speedup curves increase linearly. For example on decision tree classifier, it achieves the relative speedup of 1.185, 1.315, and 1.904on 2, 3, and 4 thread, respectively. The speedup curves tend to decrease from the linear curve at 1.797, 1.730, 1.550 and 1.311 on thread 6, 8, 10 and 12.When it accesses to decision stump classifier, it achieves the relative speedup of 1.209, 1.477, 1.548, and 2.977 on 2, 3, 4 and 6 thread, respectively. The speedup curves tend to decrease from the linear curve at 1.720, 1.661 and 1.626 on thread 8, 10 and 12.

The curve tends to stability when ID3 Numerical algorithm is applied and it achieves the speedup of 1.008, 1.024, 1.034, 1.405 and 1.469 on thread 2, 3, 6, 8 and 12.

When we go to increase the number of threads further, the speedup curves tend to drop significantly from the linear curve. For a given problem instant, the speedup saturates as the overheads grow with increasing the number of threads.

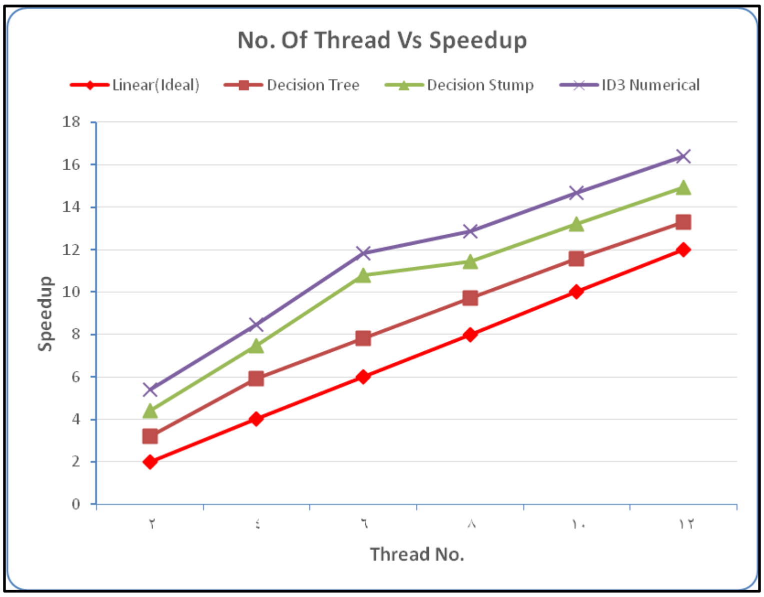


Figure 5.12: BHA Relative Speedup curve for the parallel classifiers

In addition, we compute the efficiency, which gained from this parallelization. The efficiency recorded in Table 5.6. Figure 5.13 illustrates the efficiency curves.

Table 5.6: BHA Relative Efficiency of the Parallel Classifiers

Parallel Algorithm Thread No	Decision Tree	Decision Stump	ID3 Numerical
2	0.592	0.605	0.504
3	0.438	0.492	0.341
4	0.476	0.387	0.256
6	0.300	0.496	0.172
8	0.216	0.215	0.176
10	0.155	0.166	0.145
12	0.109	0.136	0.122

We note from Table 5.6, that the value of efficiency is between zero and one. We note that the efficiency decreases as the number of threads increased for a given problem and this is common to all parallel programs due to increased overheads.

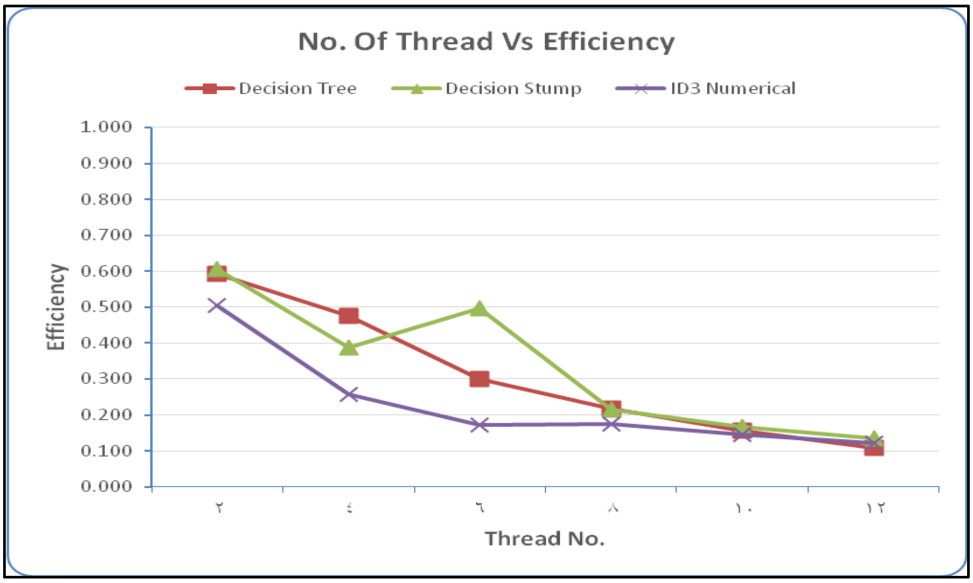


Figure 5.13: BHA Relative Efficiency curve for the parallel classifiers

In addition, we compute the parallel overhead. The parallel overhead values registered in Table 5.7. Figure 5.14 illustrates the parallel overhead curves.

Table 5.7: BHA Relative Overhead of the Parallel Classifiers

Parallel Algorithm Thread No	Decision Tree	Decision Stump	ID3 Numerical
2	0.688	0.654	0.983
3	1.281	1.031	1.931
4	1.101	1.584	2.906
6	2.338	1.016	4.805
8	3.624	3.651	4.696
10	5.451	5.021	5.898
12	8.151	6.380	7.171

As we note from Table 5.7, the parallel overhead of the parallel classifiers increases as we increase the number of threads for a given problem. This is a normal situation when the problem size fixed as the number of threads increases.

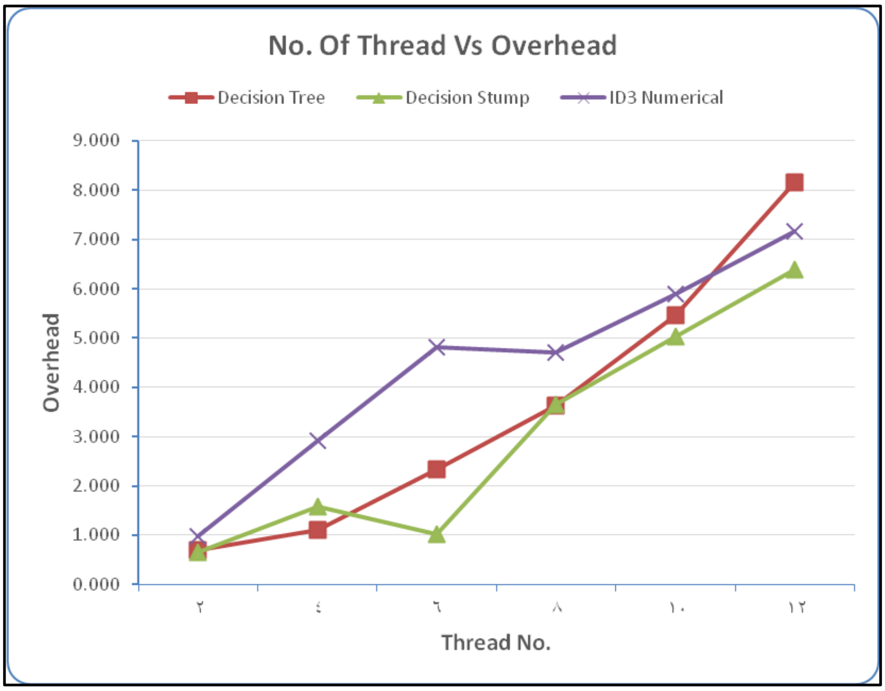


Figure 5.14: BHA Relative Overhead curve for the parallel classifiers

From figures 5.11, 5.12, 5.13,514 we can conclude that the best classifier is decision tree with 90.63% accuracy by using four threads, with an acceptable time of 219 Sec, overhead 1.101, speedup 1.904, and efficiency 0.476.

5.5 OCA Parallel Opinion mining

We apply parallel opinion mining on the other corpus which is OCA, also we apply the family of parallel decision tree classification algorithms to observe the effects of different classifiers on the performance, we also varied the number of threads from 2 to 12, and we repeated each experiment five times to get the average accuracy, recall, precision, and f-measure which recorded in table 5.8.

Table 5.8: OCA Parallel Accuracy and F-Measure

Algorithm	THREAD NO	Accuracy	Recall	Precision	F-Measure
Decision Tree (Parallel)	2	92.59	92.58	92.58	92.58
	3	91.36	91.30	93.86	92.56
	4	**93.83**	93.86	92.58	**93.22**
	6	92.59	92.58	91.45	92.01
	8	91.36	91.30	92.58	91.94
	10	88.89	88.83	91.44	90.12
	12	92.59	92.58	88.94	90.72
Decision Stump (Parallel)	2	77.78	76.92	85.00	80.76
	3	77.78	76.92	85.00	80.76
	4	77.78	76.92	85.00	80.76
	6	77.78	76.92	85.00	80.76
	8	74.08	73.17	80.87	76.83
	10	67.90	67.95	64.69	66.28
	12	77.78	76.92	85.00	80.76
ID3 Numerical (Parallel)	2	**93.83**	93.77	93.94	**93.85**
	3	92.59	92.58	92.58	92.58
	4	92.59	92.58	92.75	92.66
	6	92.59	92.58	92.58	92.58
	8	92.59	92.58	92.58	92.58
	10	92.59	92.58	92.58	92.58
	12	92.59	92.58	92.58	92.58

From table 5.8 we note that in the case of applying parallel decision tree we get the best results of accuracy (93.83%) and f-measure (93.22) at thread 4, which is greater than sequential that have an accuracy of (92.59%) and f-measure (92.58), in the case of applying parallel decision stump the accuracy (77.78%) and f-measure (80.76) which is mostly the same as sequential accuracy is of (77.78%) and f-measure (80.76), and in the case of applying parallel ID3 the accuracy (93.83%) and f-measure (93.85) at thread 2 it is greater than sequential accuracy (92.59%) and f-measure (92.58).The results of accuracy are illustrated in figure 5.15.

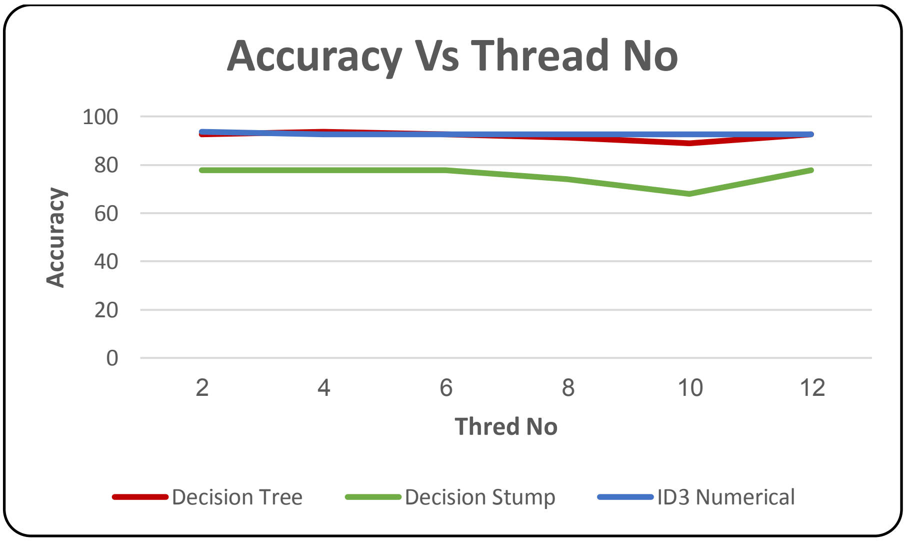


Figure 5.15: OCA Relative Accuracy curve for the parallel classifiers

The execution time in seconds recorded in Table 5.9

Table 5.9: OCA Parallel execution time

THREAD NO	Decision Tree Time (sec)	Decision Stump Time (sec)	ID3 Time(sec)
2	57.00	06.00	42.00
3	49.00	07.00	44.00
4	42.00	05.00	42.00
6	50.00	04.00	46.00
8	52.00	05.00	47.00
10	55.00	07.00	49.00
12	57.00	08.00	51.00

When we pay attention to the above Table 5.9 we can make several observations:

First, parallel Decision Tree classifier consumed the lowest time 42 seconds and gained the highest accuracy 93.83 by using 4 threads, but consumed the highest time 57.00 by using 2 and 12 threads, However, the parallel achieves a good execution time compared to sequential one. Also we note that the execution time decreases when the number of threads increases to reach 4 but increase when the number of threads increases from 6 to 12.

We note from Figure 5.16 that the sequential Decision Tree classifier spent a lot of time which is 68 seconds to classify the text documents, and the parallel version clearly reduces this time to 42 seconds on 4 threads.

Second, parallel Decision Stump classifier consumed the lowest time 4 seconds by using 6 threads, but consumed the highest time 8.00 by using 12 threads, However, the parallel achieves a good execution time compared to sequential one. Also we note that the execution time decreases when the number of threads increases to reach 6 but it increases when the number of threads increases from 8 to 12.

We note from Figure 5.16 that the sequential Decision stump classifier spent a lot of time which is 8 seconds to classify the text documents, and the parallel version of it clearly reduces this time to 4 seconds on 6 threads.

Third, parallel ID3 numerical classifier consumed the lowest time 42 seconds and gained the highest accuracy 93.83 by using 2 threads, But consumed the highest time 51.00 by using 12 threads, However, the parallel achieves a good execution time compared to sequential one. In addition, we note that the execution time decreases when the number of threads increases to reach 2 but increases when the number of threads increases from 3 to 12.

We note from Figure 5.16 that the sequential ID3 classifier spent a lot of time (65 seconds) to classify the text documents, and the parallel version clearly reduces this time to 42 seconds on 2 threads.

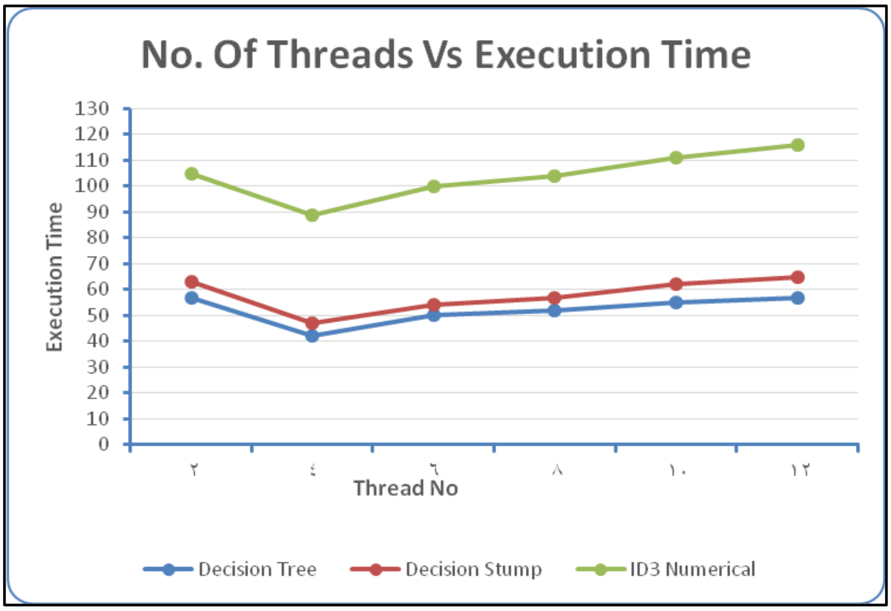


Figure 5.16: The curves of execution time for the parallel classifier for OCA

From the parallel and sequential consumed time we computed the speedup and recorded the results in table 5.10

Table 5.10: OCA Relative Speedup of the Parallel Classifiers

Parallel Algorithm	Decision Tree	Decision Stump	Id3 Numerical
Thread No			
2	1.19	1.33	1.55
3	1.39	1.14	1.48
4	1.62	1.60	1.55
6	1.36	2.00	1.41
8	1.31	1.60	1.38
10	1.24	1.14	1.33
12	1.19	1.00	1.27

By analyzing the results in the above table we found that Speedup achieved by parallel decision tree classifier increased linearly while the thread number increased to hit 1.62 at thread 4 but it linearly decreased to hit 1.19 at thread 12. And from figure 5.17 we clearly demonstrated that the speedup curves increase linearly. For example on decision tree classifier it achieves the relative speedup of 1.19, 1.39, and 1.62 on 2, 3 and 4 thread, respectively. Speedup curves tend to decrease from the linear curve at 1.36, 1.31, 1.24 and 1.19 on thread 6, 8, 10 and 12.

Also, we found that the speedup achieved by parallel decision stump classifier increased while the thread number increase to hit 2.00 at thread 6 but linearly decreased to hit 1.00 at thread 12. And from figure 5.17 we clearly demonstrated that the speedup curves increases. For example on decision stump classifier, it achieves the relative speedup of 1.14, 1.60, and 2.00 on 3, 4 and 6 thread, respectively. The speedup curves tend to decrease from the linear curve at 1.60, 1.14 and 1.00 on thread 8, 10 and 12.

Furthermore, we found that the highest speedup achieved by parallel ID3 classifier is of 1.55 on thread 2 and 4. From figure 5.17 we clearly demonstrated that the speedup curves tend to decrease at 1.41, 1.38, 1.33 and 1.27 on thread 6, 8, 10 and 12.

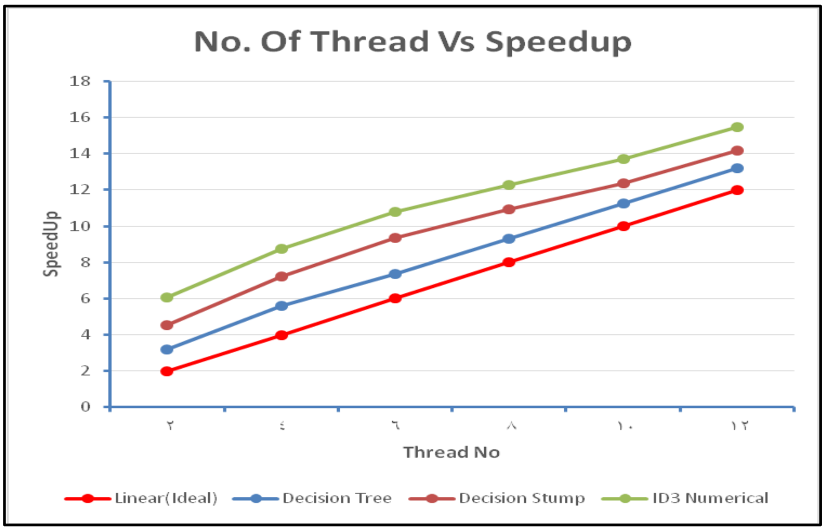


Figure 5.17: The curves of Speedup for the parallel classifier for OCA

In addition, we were able to calculate the efficiency, which gained from the speedup and the number of threads. The calculated efficiency registered in Table 5.11. Figure 5.18 illustrates the efficiency curves.

Table 5.11: OCA Relative Efficiency of the Parallel Classifiers

Parallel Algorithm Thread No	Decision Tree	Decision Stump	Id3 Numerical
2	0.60	0.67	0.77
3	0.46	0.38	0.49
4	0.40	0.40	0.39
6	0.23	0.33	0.24
8	0.16	0.20	0.17
10	0.12	0.11	0.13
12	0.10	0.08	0.11

We deduce from Table 5.11, that the value of efficiency is between zero and one. Moreover, this clearly demonstrates that the efficiency decreases as the number of threads increase for a given problem and this is common to all parallel programs due to increased overheads.

From figure 5.18 we found that the value of efficiency decreases linearly as the number of thread increases linearly.

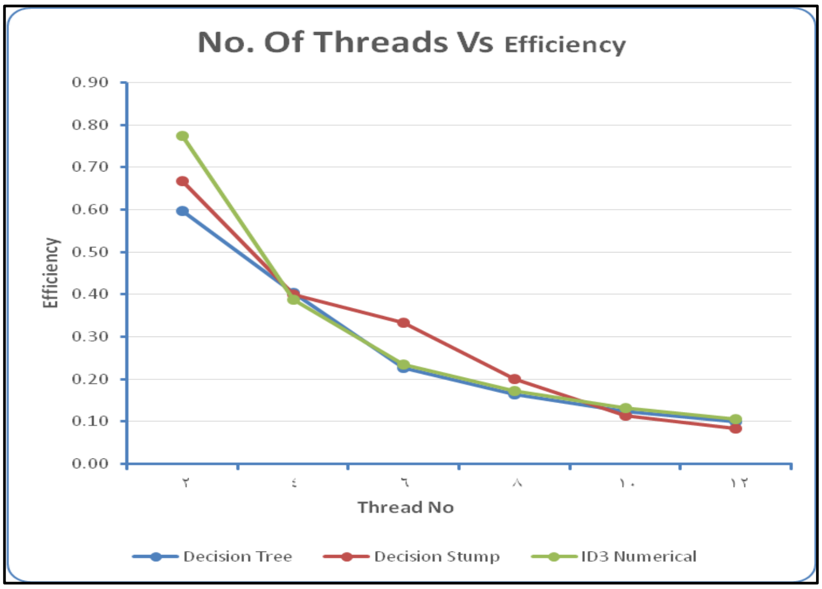


Figure 5.18: The curves of Efficiency for the parallel classifier for OCA

In addition, we went to compute the time spent by all processors combined in non-useful work, called the parallel overhead, we recorded these values in Table 5.12. Figure 5.19 illustrates the parallel overhead curves.

Table 5.12: OCA Relative Overhead of the Parallel Classifiers

Parallel Algorithm Thread No	Decision Tree	Decision Stump	ID3 Numerical
2	0.68	0.50	0.29
3	1.16	1.63	1.03
4	1.47	1.50	1.58
6	3.41	2.00	3.25
8	5.12	4.00	4.78
10	7.09	7.75	6.54
12	9.06	11.00	8.42

When we examined table 5.12 and figure 5.19 we found that the parallel overhead fits directly proportional to the number of threads as it increases when the thread number increases. The best case of overhead is of 8.42 when applied parallel ID3 classifier, the worst case is 11.00 when applied parallel decision stump, and the average is 9.06 when applied parallel decision tree classifier.

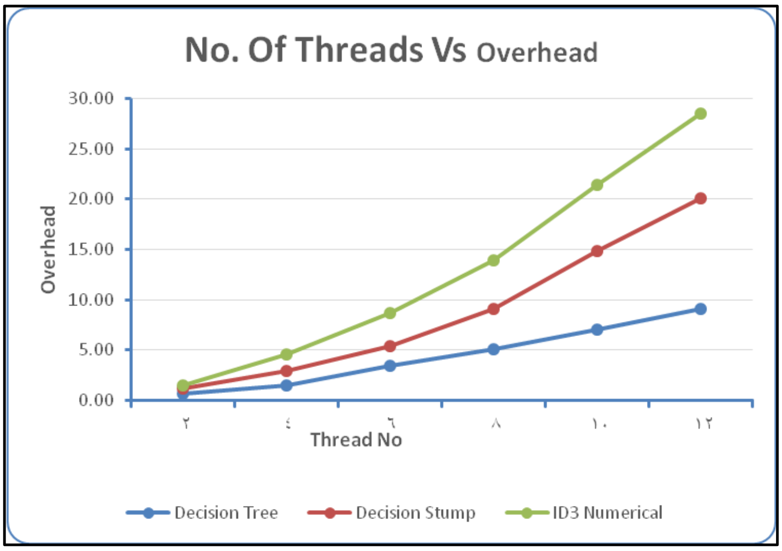


Figure 5.19: The curves of Overhead for the parallel classifier for OCA

From figure 5.16, 5.17, 5.18, 5.19 we can conclude that the ID3 and decision tree classifiers have the same accuracy 93.83% and consumed time 42 Sec, ID3 get this time by using 2 threads but decision tree with 4, ID3 used less number of threads. In the efficiency viewpoint we found that, ID3 is the best as its efficiency is of 0.77 but Decision tree efficiency is of 0.40.

5.6 Summary

In this chapter, we applied a family of decision tree learning algorithms such as decision tree, Quinlan's ID3 numerical, and decision stumps on OCA and BHA Arabic corpuses. In addition, we presented and analyzed the experimented results after learning OCA and BHA corpus by the sequential and parallel decision trees learning algorithms.

Chapter 6

Conclusion and Future Works

This chapter draws a conclusion, which includes its results; discussion and comparing decision trees with other classification methods, and then gives some suggestions for future works.

6.1 Conclusion

People’s opinion becomes an essential part of our information-collection behavior. In order to make decision there has been a growing interest in the automatic detection of opinionated content in natural language text. Arabic Opinion mining can be seen as classification problem where documents are classified as positive or negative. One of the common classification algorithms is decision tree, which give us more understandable results about why results appear so. However, the Decision tree algorithm is of low efficiency when it is used to handle a large volume of text documents with high dimensionality and in particular in the Arabic language large. So that we used parallel decision tree methods.

For our experiments, we applied parallel decision tree on two Arabic corpuses of text documents by using RapidMiner environment [87], the first corpus is Booking of hotels for Arabic (BHA) that is collected from online Arabic economic websites, including tripadvisor.com.eg , booking.com, and agoda.ae which has reviews about hotels, resorts, flights, vacation, travel packages, and lots more. To generate text representations for BHA corpus we applied five operators: Tokenize , filters Arabic stopwords, Stem Arabic, filters tokens based on their length, and filters tokens based on their content to exclude English words. The second corpus is Opinion corpus for Arabic (OCA), which is a corpus of text from movie review sites by Saleh et al. [6]. To generate text representations for OCA corpus we applied three operators: Tokenize, filters Arabic stopwords, and filters tokens based on their content to exclude English words.

In this research, we effectively handle both Arabic corpuses by using a family of parallel decision tree to classify documents as positive or negative and to get more understandable and efficient results about why results appear so. To observe the effects of different classifier on the performance, we varied the number of threads from 2 to 12, and we repeated each experiment five times to get the average accuracy, recall, precision, and f-measure.

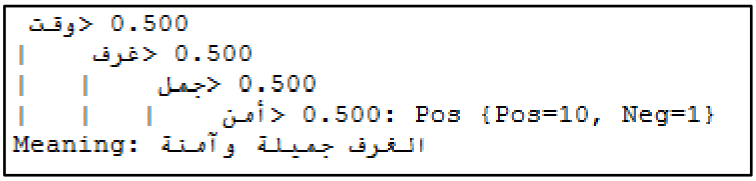
For evaluation purposes, we used Accuracy, Precision, Recall, F-Measure, Speedup, Efficiency, and Overhead.

In case of applying parallel decision tree family on OCA we get the best results of accuracy (93.83%) and f-measure (93.22) at thread 4, which is greater than sequential that have accuracy (92.59%) and f-measure (92.58), and consumed time 42 Sec by using decision tree and ID3 classifiers. Also the highest speedup 1.62 is achieved by decision tree classifier using 4 threads. The best case of overhead is 8.42 when applied parallel ID3 classifier, the worst case is 11.00 when applied parallel decision stump, and the average is 9.06 when applied parallel decision tree classifier.

In case of applying parallel decision tree family on BHA we get the best results of accuracy (90.63%) and f-measure (82.29) at thread 4, these results are different from sequential that have accuracy (90.70%) and f-measure (90.94), and consumed time 219 Sec, and highest speed up1.904 by using decision tree. Nevertheless, ID3 gives us 90.7 accuracy in 2300 sec. The best case of overhead is 6.380 when applied parallel decision stump classifier, the worst case is 8.151 when applied parallel decision tree, and the average is 7.171 when applied parallel ID3 classifier.

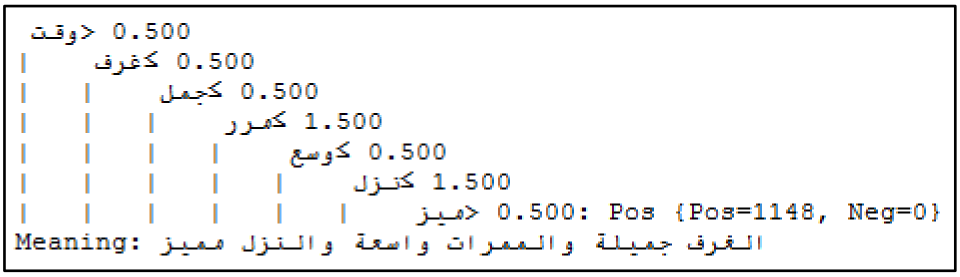
Some of the resulted rules from BHA:

These are some of reasons or rules why people's opinion is positive about the hotels, we explained it in the following figures.



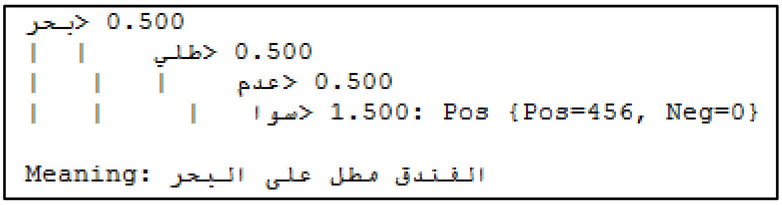
```
 وقت> 0.500
|    غرف> 0.500
|    |    جمل> 0.500
|    |    |    أمن> 0.500: Pos {Pos=10, Neg=1}
Meaning: الغرف جميلة وآمنة
```

Figure 6.1: Rule 1 from BHA positive opinions



```
 وقت> 0.500
|    كغرف 0.500
|    |    كجمل 0.500
|    |    |    كمرر 1.500
|    |    |    |    كوسع 0.500
|    |    |    |    |    كنزل 1.500
|    |    |    |    |    |    ميز> 0.500: Pos {Pos=1148, Neg=0}
Meaning: الغرف جميلة والممرات واسعة والنزل مميز
```

Figure 6.2: Rule 2 from BHA positive opinions



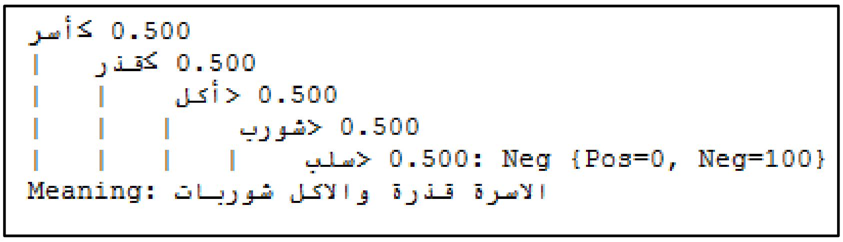
```
بحر> 0.500
|  |    طلي> 0.500
|    |    |    عدم> 0.500
|    |      |  سوا> 1.500: Pos {Pos=456, Neg=0}

Meaning: الفندق مطل على البحر
```

Figure 6.3: Rule 3 from BHA positive opinions

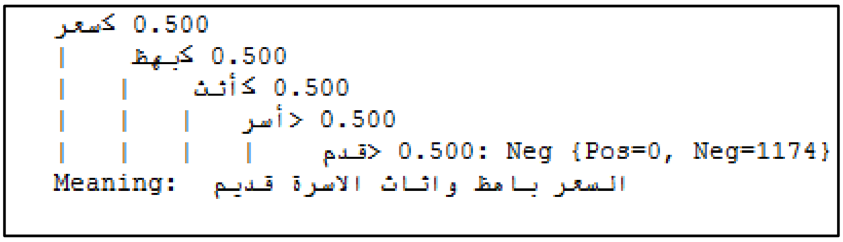
From the above figures 6.1, 6.2, 6.3 we can extract understandable and clear knowledge about positive people's opinion, This knowledge where credit is due to the size of BHA corpus which is 6934 examples.

These are some of reasons why people's opinion is negative about the hotels, we explained it in the following figures.



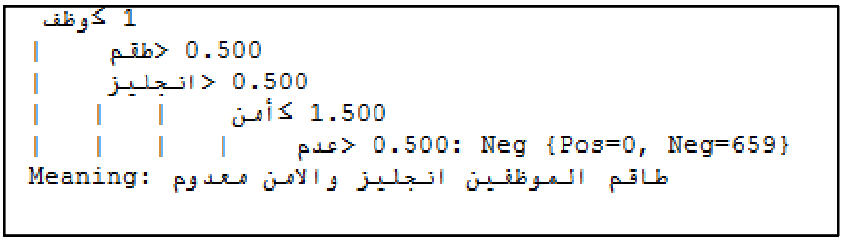
```
أسر ≤ 0.500
|   قذر ≤ 0.500
|   |   أكل > 0.500
|   |   |   شورب > 0.500
|   |   |   |   سلب > 0.500: Neg {Pos=0, Neg=100}
Meaning: الاسرة قذرة و الاكل شوربات
```

Figure 6.4: Rule 1 from BHA negative opinions



```
  سعر ≤ 0.500
  |   بهظ ≤ 0.500
  |   |   أثث ≤ 0.500
  |   |   |   أسر > 0.500
  |   |   |   |   قدم > 0.500: Neg {Pos=0, Neg=1174}
  Meaning:  السعر باهظ واثاث الاسرة قديم
```

Figure 6.5: 8 Rule 2 from BHA negative opinions



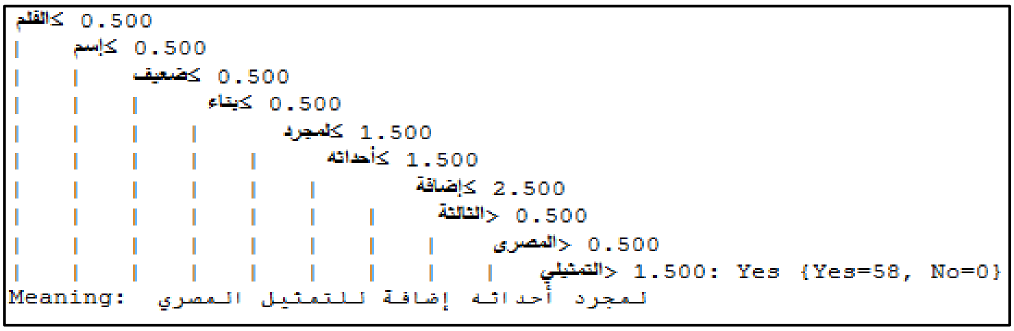
```
 وظف ≤ 1
|   طقم > 0.500
|   انجليز > 0.500
|   |   |   أمن ≤ 1.500
|   |   |   |   عدم > 0.500: Neg {Pos=0, Neg=659}
Meaning: طاقم الموظفين انجليز والامن معدوم
```

Figure 6.6: Rule 3 from BHA negative opinions

From the above figures 6.4, 6.5, 6.6 we can extract understandable and clear knowledge about negative people's opinion about the hotels.

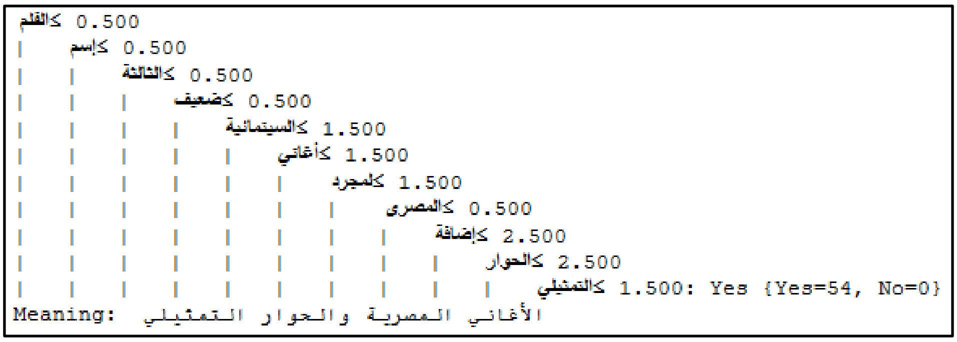
Some of the resulted rules from OCA:

These are some of reasons why people's opinion is positive about the film, we explained it in the following figures.



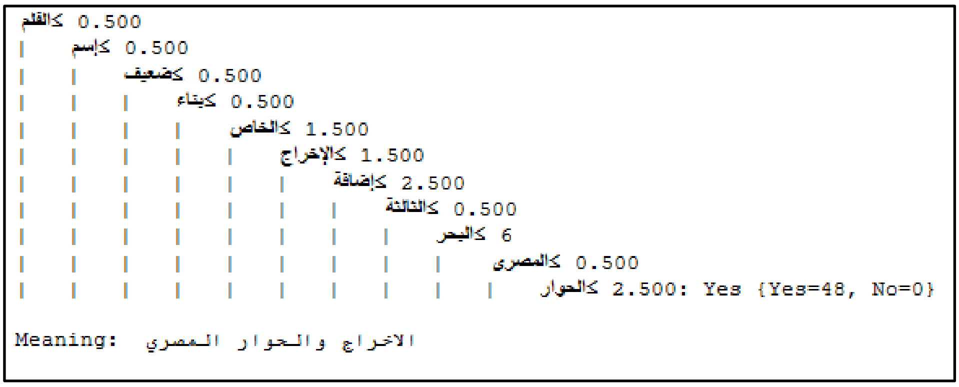
```
القلم ≤ 0.500
|   إسم ≤ 0.500
|   |   ضعيف ≤ 0.500
|   |   |   بناء ≤ 0.500
|   |   |   |   لمجرد ≤ 1.500
|   |   |   |   |   أحداثه ≤ 1.500
|   |   |   |   |   |   إضافة ≤ 2.500
|   |   |   |   |   |   |   الثالثة > 0.500
|   |   |   |   |   |   |   |   المصري > 0.500
|   |   |   |   |   |   |   |   |   التمثيلي > 1.500: Yes {Yes=58, No=0}
Meaning:  لمجرد أحداثه إضافة للتمثيل المصري
```

Figure 6.7: Rule 1 from OCA positive opinions



```
القلم ≤ 0.500
|    إسم ≤ 0.500
|    |    الثالثة ≤ 0.500
|    |    |    ضعيف ≤ 0.500
|    |    |    |    السيتمانية ≤ 1.500
|    |    |    |    |    أغاني ≤ 1.500
|    |    |    |    |    |    لمجرد ≤ 1.500
|    |    |    |    |    |    |    المصرى ≤ 0.500
|    |    |    |    |    |    |    |    إضافة ≤ 2.500
|    |    |    |    |    |    |    |    |    الحوار ≤ 2.500
|    |    |    |    |    |    |    |    |    |    التمثيلي ≤ 1.500: Yes {Yes=54, No=0}
Meaning:  الأغاني المصرية والحوار التمثيلي
```

Figure 6.8: Rule 2 from OCA positive opinions



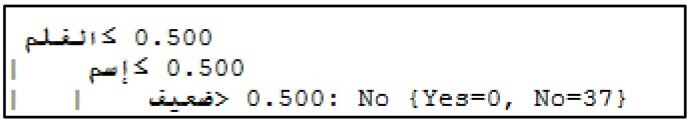
```
القلم ≤ 0.500
|    إسم ≤ 0.500
|    |    ضعيف ≤ 0.500
|    |    |    بناء ≤ 0.500
|    |    |    |    الخاص ≤ 1.500
|    |    |    |    |    الإخراج ≤ 1.500
|    |    |    |    |    |    إضافة ≤ 2.500
|    |    |    |    |    |    |    الثالثة ≤ 0.500
|    |    |    |    |    |    |    |    البحر ≤ 6
|    |    |    |    |    |    |    |    |    المصري ≤ 0.500
|    |    |    |    |    |    |    |    |    |    الحوار ≤ 2.500: Yes {Yes=48, No=0}

Meaning:  الاخراج والحوار المصري
```

Figure 6.9: Rule 3 from OCA positive opinions

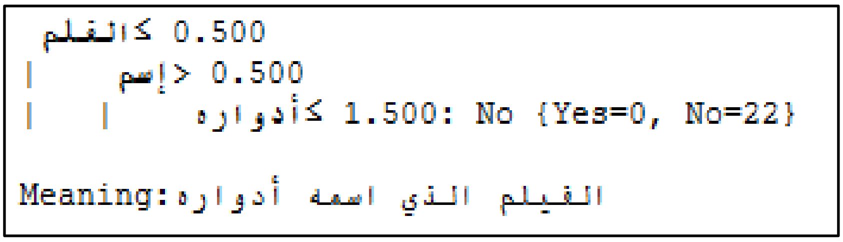
From the above figures 6.7, 6.8, 6.9 we can extract knowledge about positive people's opinion about the films.

These are some of reasons why people's opinion is negative about the film, we explained it in the following figures.



```
القلم ≤ 0.500
|    إسم ≤ 0.500
|    |    ضعيف > 0.500: No {Yes=0, No=37}
```

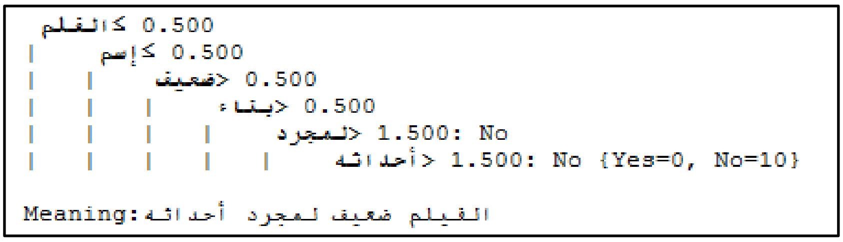
Figure 6.10: Rule 1 from OCA negative opinions



```
القلم ≤ 0.500
|    إسم > 0.500
|    |    أدواره ≤ 1.500: No {Yes=0, No=22}

Meaning:الفيلم الذي اسمه أدواره
```

Figure 6.11: Rule 2 from OCA negative opinions



```
 الفلم≤ 0.500
|    إسم≤ 0.500
|    |    ضعيف> 0.500
|    |    |    بناء> 0.500
|    |    |    |    لمجرد> 1.500: No
|    |    |    |    |    أحداثه> 1.500: No {Yes=0, No=10}

Meaning:أحداثه لمجرد ضعيف الفيلم
```

Figure 6.12: Rule 3 from OCA negative opinions

From the above figures 6.10, 6.11, 6.12 we can extract knowledge about negative people's opinion about the films, we noted that we can't get knowledge from figure 6.10 due to the size of OCA corpus which is 135 examples.

6.2 Future Works

In the future works,

we will generalize our work to other kinds user generated contents such as Internet forums, discussion groups, blogs , log files, and social networks. Also, we can generalize it to other application domains such as distance transformation points, traffic, , and medical information and Educational data. Our work can be extended to cover larger computer clusters and text corpora. Also we will extend our work to cover a popular distributed programming paradigm like Map Reduce in a cloud environment. In addition we can use other types of data such as multimedia Data.

References

[1] Aas K., and Eikvil L., “Text Categorization: A survey”, Technical report, Norwegian Computing Center, 1999.

[2] Abdelali, A., and Cowie, J., “Regional corpus of modern standard Arabic”, In 2nd Int. Conf. on Arabic Language Engineering, 2005, Vol. 1, No. 1, 2005, pp. 1-12.

[3] Abdul-Mageed M. , and Diab M. , “Subjectivity and Sentiment Annotation of Modern Arabic Newswire”. In Proceedings of the 5th Law Workshop (LAW V). Portland, Organ, 2011

[4] Abu Tair M., “A High Performance Parallel Classifier for Large-Scale Arabic Text”, M.Sc. Dissertation, Department of Information Technology, The Islamic University-Gaza.

[5] Almas Y. ,and Ahmad K. ,” A Note on Extracting Sentiments in Financial News in English, Arabic & Urdu”, In Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages Linguistic Institute, Stanford, California, USA, 2007.

[6] Ameed H., Al-Ketbi Sh., Al-Kaabi A., Al-Shebli Kh., Al-Shamsi N., Al-Nuaimi N., Al-Muhairi Sh., “Arabic Light Stemmer: A New Enhanced Approach,” in Proceedings of Software Engineering Department, UAE University, Dubai, 2005,pp. 1-9

[7] Asaleem S.,“Automated Arabic Text Categorization Using SVM and NB”, The International Arab Journal of e-Technology, vol.2, no.2, 2011

[8] Azara M., Fatayer T., El-Halees A., “Arabic Text Classification Using Learning Vector Quantization”, The 8th International Conference on INFOrmatics and Systems (INFOS2012) – 14-16 May Natural Language Processing Track, Faculty of Computers and Information - Cairo University, pp.NLP-40

[9] Ben-Haim Y. and Tom-Tov E..” A Streaming Parallel Decision Tree Algorithm”. Journal of Machine Learning Research, vol. 11, 2010, pp. 849-872.

[10] Biao Q. , Yuni X. , Sunil P.,and Yicheng T. , “A Rule-Based Classification Algorithm for Uncertain Data”, IEEE International Conference on Data Engineering, 2009

[11] Bing L., “Mining and Searching Opinions in User-Generated Content on the Web”, Invited talk at the Sixth Annual Emerging Information Technology Conference , Dallas, Texas, 2006.

[12] Bing L., “Opinion Mining”, Morgan & Claypool Publishers, 2012.

[13] Bing L., “Sentiment Analysis and Opinion Mining”, Morgan & Claypool Publishers, vol. 15,no.184, 2012, pp.1-2

[14] Chen A., “Building an Arabic Stemmer for Information Retrieval,” in Proceedings of the Eleventh Text Retrieval Conference, Berkeley, 2003, pp. 631-639

[15] Ciravegna F., Gilardoni L., Lavelli A., Ferraro M., Mana N., Mazza, S., Matiasek J., Black W., and Rinaldi F., “Flexible Text Classification for Financial Applications: the FACILE System”, In Proceedings of PAIS-2000, Prestigious Applications of Intelligent Systems sub-conf. of ECAI2000, 2000.

[16] Darwish K., “Building a shallow Arabic morphological analyzer in one day”. In Proceedings. Of the 40th Annual Meeting of the Association for Computational Linguistics (ACL’02), 2002

[17] Darwish K., “Probabilistic methods for searching OCR-degraded Arabic text”, PhD thesis, University of Maryland, College Park, Maryland, United States, 2003

[18] Dave D., Lawrence A., and Pennock, D. “Mining the Peanut Gallery: Opinion Extraction andSemantic Classification of Product Reviews”. Proceedings of International World Wide Web Conference (WWW’03), 2003

[19] Dunham M. , “Data Mining: Introductory and Advanced Topics”. 1st Edition, Pearson Education, 2003.

[20] Duwairi R., “A Distance-based Classifier for Arabic Text Categorization”, In the Proc. of the Int. Conf. on Data Mining, Las Vegas, USA, 2005.

[21] Duwairi R., “Arabic text Categorization”, In the Int. Arab journal of information technology, 4(2), 2007.

[22] Duwairi R., “Arabic Text Categorization”, The International Arab Journal of Information Technology, vol.4, no.2, 2007

[23] Duwairi R., “Machine Learning for Arabic text Categorization”, Journal of the American Society for Information Science and Technology, vol.57, no. 8, pp. 1005-1010. 2006.

[24] Duwairi R., Al-Refai M., Khasawneh N., “Feature reduction techniques for Arabic text categorization”, Journal of the American Society for Information Science, vol. 60, no. 11, pp. 2347-2352, 2009.

[25] Duwairi R., Al-Refai M., Khasawneh N., “Stemming Versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization”, In the 4th Int. Conf. of Innovations in Information Technology, IIT'07, pp. 446 – 450, 2007.

[26] Duwairi, R., Al-Refai, M., and Khasawneh, N. , “Feature reduction techniques for Arabic text categorization”, Journal of the American Society for Information Science, vol. 60, no. 11,2009, pp. 2347-2352.

[27] Edward L., Steven B., and Klein E. ,”Natural language processing with Python”. Sebastopol, CA: O'Reilly. ISBN 0-596-51649-5,2009

[28] Elarnaoty M., AbdelRahman S., and Fahmy S., “A Machine Learning Approach For Opinion Holder Extraction In Arabic Language”, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March ,2012.

[29] El-Halees A. , “Arabic Opinion Mining Using Combined Classification Approach”. Proceeding The International Arab Conference On Information Technology, Azrqa, Jordan.,2011.

[30] El-Halees A., “A Comparative Study on Arabic Text Classification”, Egyptian Computer Science Journal, vol.20, no.2, 2008.

[31] Elhawary M. ,and Elfeky M. ,” Mining Arabic Business Reviews”, IEEE International Conference on Data Mining Workshops, 2010.

[32] Farra N., Challita E., Abou-Assi R, and Hajj H., “Sentence-level and Document-level Sentiment Mining for Arabic Texts”, 2010 IEEE International Conference on Data Mining Workshops,2010.

[33] Feldman R., Sanger J., “The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data”, Cambridge University Press, 2007.

[34] Fernande J., “Prediction of protein binding sites and hot spots”, Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 1, no. 5, 2011, pp. 680-698, Available: http://onlinelibrary.wiley.com/doi/10.1002/wcms.45/abstract

[35] Ghosh S., Roy S., and Bandy S., “A tutorial review on Text Mining Algorithms”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, no. 4, June 2012
Grama A., Gupta A., Karypis A., and Kumar V., “Introduction to Parallel Computing”, 2nd edition, Addison Wesley,2003.

[36] Grama, A., Gupta, A., Karypis, G. and Kumar, V. “Introduction to Parallel Computing”, 2nd edition, Addison Wesley,2003.

[37] Harrag F., El-Qawasmeh E., and Pichappan P., “Improving Arabic text categorization using decision trees”, In the 1st Int. Conf. of NDT '09, pp. 110 – 115, 2009.

[38] Hill T., Lewicki P.,“STATISTICS Methods and Applications”, (1st Ed), StatSoft, Tulsa, OK, 2007

[39] http://sinai.ujaen.es/wiki/index.php/OCA_Corpus_(English_version)

[40] Janyce W., and Riloff E., “Creating subjective and objective sentence classifiers from unannotated texts”, Computational Linguistics and Intelligent Text Processing, 2005, pp. 486–497

[41] Jindal N. , and Bing L. “Mining comparative sentences and relations”, In Proceedings of National Conf. on Artificial Intelligence (AAAI-2006).2006.

[42] Joachims T., “Text categorization with support vector machines: Learning with many relevant features,” in Proceedings of the 10th European Conference on Machine Learning, 1998, pp. 137-142.

[43] Joshi M., Karypis G. , and Kumar, V. “ScalParC : A New Scalable and Efficient Parallel Classification Algorithm for Mining Large Datasets ,” in Proceedings of the First Merged International and Symposium on Parallel and Distributed Processing, Orlando. FL, 1998, pp. 573 - 579

[44] Juling D., Zhongjian L, Ping Z., Gensheng W., and Wei S., “An Opinion-Tree based Flexible Opinion Mining Model”, Proceeding of 2009 International Conference on Web Information Systems and Mining, 2009.

[45] Kanchan S., and Vineet R. ,“A Survey-Classifier Fusion,” Journal of Global Research in Computer Science(JGRCS),vol. 3, no. 2, 2012, pp.25-28

[46] Khoja S., Garside R., “Stemming Arabic text”, Computer Science Department, Lancaster University, Lancaster, UK, 1999.

[47] Khreisat L., “A machine learning approach for Arabic text classification using N-gram frequency statistics”, Journal of Informetrics, Elsevier, vol.3, no.1,2009, pp. 72-77.

[48] Kohavi R., and Provost F., “Glossary of Terms Special Issue on Applications of Machine Learning and the Knowledge Discovery Process”, Kluwer Academic Publishers Hingham, Machine Learning, vol. 30, no. 2-3, 1998

[49] Kun L., “An Insight Into Vector Space Modeling and Language Modeling”, iConference, Fort Worth, TX, USA,2013

[50] Mierswa I., Wurst M., Klinkenberg R., Scholz M., Euler T., “YALE: Rapid Prototyping for Complex Data Mining Tasks”, in the Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining KDD-06, 2006.

[51] Minqing H. and Bing L. ,“Mining and summarizing customer reviews”. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004). 2004.

[52] Mitchell T., and Mcgraw H. , “Machine Learning”, January, 2010

[53] Olcay Y. ,and Dikmen O., “Parallel univariate decision trees”, Pattern Recognition Letters, vol. 28 ,2007,pp. 825–832

[54] Padraig C. , and Sarah J. , “k-Nearest Neighbour Classifiers,” Technical Report UCD-CSI-2007-4 , 2007.

[55] Pang B. and Lillian L., “Opinion Mining and Sentiment Analysis”, In Information Retrieval Vol.2, No 1-2, 2008, pp.21–135

[56] Pang B., Lillian L., and Shivakumar V.,” Thumbs up?: sentiment classification using machine learning techniques”. in Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002). 2002.

[57] Peng F., Huang X., Schuurmans D., and Wang S., “Text Classification in Asian Languages without Word Segmentation”, In Proc. of the 6th Int. Workshop on Information Retrieval with Asian Languages (IRAL 2003), Association for Computational Linguistics, Sapporo, Japan, 2003.

[58] Phyu Th., “Survey of Classification Techniques in Data Mining”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 vol I, IMECS 2009, , Hong Kong,2009

[59] Quinlan J. , “Discovering rules by induction from large collections of examples”, Expert Systems in the Micro Electronic Age, Edinburgh University Press, 1979, pp. 168–201.

[60] Quinlan J. , “Induction of Decision Trees”, Mach. Learn, vol.1, no.1 , 1986, pp.81-106

[61] Quinlan J. , “C4. 5: programs for machine learning, ”. Morgan Kaufmann, San Mateo, SA, 1993.

[62] Quinlan J. , “Improved use of continuous attributes in c4.5, ”. Journal of Artificial Intelligence Research, vol. 4, 1996, pp.77-90

[63] Ranka S.,“Classification”, Computer and Information Science and Engineering, University of Florida, Gainesville, 2011, available: http://www.cise.ufl.edu/~asaha/book.pdf

[64] Rapid I., 2010, Available: http://rapid-i.com/wiki/index.php?title=Decision_Tree

[65] Rapid I., 2010, Available: http://rapid-i.com/wiki/index.php?title=ID3

[66] Rapid I.,2010, Available: http://rapid-i.com/wiki/index.php?title=Decision_Stump

[67] Rapid I.,2010, Available: http://rapid-i.com/wiki/index.php?title=Text:Tokenize

[68] Rapid I.,2010, Available: http://rapid-i.com/wiki/index.php?title=Process_Documents_from_Files

[69] Robert C. ,“Very Simple Classification Rules Perform Well on Most Commonly Used Datasets,” Springer. Machine Learning, vol. 11, no. 1 , 1993, pp.63 – 90

[70] Rushdi-Saleh M. ,” Oca: Opinion corpus for Arabic”, Journal of the American Society for Information Science and Technology,vol. 62, no. 10, 2011, pp.2045–2054

[71] Saad M., “The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification”, M.Sc. Dissertation, Department of Computer Engineering, The Islamic University-Gaza, (2010).

[72] Saad, M. “The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification”, M.Sc. Dissertation, Department of Computer Engineering, The Islamic University-Gaza, 2010

[73] Said D., Wanas N., Darwish N.,and Hegazy N., “A Study of Arabic Text preprocessing methods for Text Categorization”, In the 2nd Int. conf. on Arabic Language Resources and Tools, Cairo, Egypt, 2009.

[74] Sawalha M., and Atwell E., “Comparative Evaluation of Arabic Language Morphological Analysers and Stemmers”, Companion volume – Posters and Demonstrations, Manchester, August 2008, pp. 107–110

[75] Saxena K., Richaria V., “A Survey-Classifier Fusion”, Journal of Global Research in Computer Science(JGRCS),vol..3, no. 2, February, 2012, pp.25-28,

[76] Shannon C. , “A mathematical theory of communication, ”. Bell System Technical Journal, vol 27, 1948, pp.379–423.

[77] Soliman T., Hedar A., and Doss M.,” Mining Social Networks’ Arabic Slang Comments”, in proceedings of IADIS European Conference on Data Mining 2013 (ECDM'13),Prague,Czech Republic, 2013

[78] Steinbach T., “Introduction to Data Mining”, 2006, pp.146-148, available: http://www-users.cs.umn.edu/~kumar/dmbook/index.php

[79] Thair N. ,“Survey of Classification Techniques in Data Mining,” Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I, IMECS 2009, Hong Kong, 2009

[80] Turney D. ,”Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews”. in Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2002). 2002.

[81] Uwec, 2001, Available: http://people.uwec.edu/piercech/researchmethods/data%20collection%20methods/data%20collection%20methods.htm

[82] Valarmathi B., and Palanisamy V. , “Opinion Mining Classification Using Key Word Summarization Based on Singular Value Decomposition”, International Journal on Computer Science & Engineering, vol. 3 , no. 1, 2011, pp212-215

[83] Wayne I. , and Langley P. ,“Induction of One-Level Decision Trees, ”, in ML92: Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, San Francisco, CA: Morgan Kaufmann, 1992, pp. 233–240

[84] Weisstein E., "Zipf's Law," MathWorld{A Wolfram Web Resource, Available: http://mathworld.wolfram.com/ZipfsLaw.html, (Date Last Accessed on 27/01/2014).

[85] Wiebe J. ,Theresa W., Rebecca F., Matthew B., and Melanie M.,”Learning subjective language”. Computational Linguistics, vol.30, no.3 , 2004, pp. 277-308.

[86] Wiebe J., Rebecca F., and Thomas P. ,”O'Hara. Development and use of a gold-standard data set for subjectivity classifications”. In Proceedings of the Association for Computational Linguistics (ACL-1999). 1999.

[87] Yang T. , “Computational Verb Neural Networks”, International Journal Of Computational Cognition, vol. 5, no. 3, September, 2007, http://www.ijcc.us

[88] Yiqun C.,”Parallel and Distributed Techniques in Biomedical Engineering”, M.Sc. Dissertation, Department of Electrical Computer Engineering, National University of Singapore.

[89] Zaki M., Ching-Tien H. and Agrawal R., “Parallel Classification for Data Mining on Shared-Memory Multiprocessors”, Data Engineering, 1999. Proceedings., 15th International Conference on ,1999, pp. 198 - 205