Application of Self-Organizing Maps to Multilingual Text Mining (Arabic-English)

Ph.D. Thesis

Abdulsamad A. M. Saleh

Software Technology Research Laboratory

Faculty of Computing Sciences and Engineering

De Montfort University

April 2008

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

In The Name of Allah
The Most Gracious, The Most Merciful

Abstract

Computing systems are becoming more and more complex and are assuming more and more responsibilities in all sectors of human activity. Science and technology information present a rich resource, essential for managing research and development programs. Many of today's applications are built as distribution systems. The Internet is one of the best-known distribution systems and is used by nearly everyone today. With a great deal of available data on the net in different languages, it is essential to use efficient methods to extract useful information from the data. Fortunately, the parallel growth of information and of analytical tools offer the promise of advanced decision aids to support research and development more effectively. Data mining, information retrieval and other information-based technologies especially nowadays, are receiving increased attention.

The importance of English is well established in every field. Likewise, Arabic is also a major natural language, spoken by over 250 millions people in 21 Arab countries as the first language, and in Islamic countries it is used as a second language. It is one of the languages of the Semitic family and thus preserves the complexity of this group. Arabic is highly derivated, as well as being an inflected language, so it requires good stemming for effective text mining. Yet no standard approach to stemming has emerged. This work investigates some of the issues involved in achieving bilingual text mining from large bodies of electronic Arabic-English datasets.

The main aim of this thesis is to address the above issues and provide the best framework. To address this aim, this thesis evaluates the current proposed pre-processing and SOM clustering algorithms. Our proposed MLTextMAES approach has the ability to perform the four main stages of standard text mining, taking into account pre-processing, clustering (via SOM) and test of quality. Thus we have employed SOM as a tool for the clustering of documents into groups with similar categories.

To the author's knowledge there is no significant literature available regarding the SOM technique applied to Arabic-English text mining. The model is found to be useful in strategic decision-making settings. The results indicate that SOM is a feasible tool for multilingual languages, and presents several advantages over current methods. Our experimental results show improved clustering performance when using Arabic-English language documents for our datasets.

Declaration

I declare that this work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the School of Computing, Faculty of Computing Science and Engineering at De Montfort University, Leicester, United Kingdom.

This thesis written by me and produced using LaTeX.

AbdulSamad A. Al-Marghilani

April, 2008

List of Publications

During the course of the incremental research, the results have been reported in a number of scientific papers.

Al-Marghilani, Abdulsamad. and Zedan, Husein. and Ayesh, Aladdin. A General Framework for Multilingual Text Mining Using Self-Organizing Maps. In Proc. of the 25th IASTED International Multi-Conference Artificial Intelligence and Applications, Innsbruck, Austria. 2007, pp.549-106.

Al-Marghilani, Abdulsamad. and Zedan, Husein. and Ayesh, Aladdin. Practical Approach Using Self-Orgnaizing Maps For Multilingual Text Mining. Proceeding of the Saudi Innovation Conference. Newcastle, UK, 2007, pp. 676-685. ISBN: 978-0-955104-92-3.

Al-Marghilani, Abdulsamad. and Zedan, Husein. and Ayesh, Aladdin. Text Mining Based on Self-Organizing Map Method for Arabic-English Documents. In prec. of the 19 MAICS Artificial Intelligence Science Conference, Cincinnati, OH, USA. April 12-13, 2008.

Dedication

إهـــــــــــــداء

I would like to dedicate this thesis to my Parents and my beloved wife and to my children Ahmed, Salman, Asma, and Somiyah.

Acknowledgments

Firstly, I thank God for giving me the ability to complete this research and to learn a little more about one small aspect of His universe.

Studying at the University of De Montfort in Leicester and particularly the School of Computing & Engineering was the most rewarding experience I have ever had. To me, this is undoubtedly related to the high standard of the facilities offered to students, the friendly atmosphere among the research students, and above all the excellence of supervision.

For this reason I would like first to express my deepest appreciation and lasting gratitude to my supervisor Prof. Hussein Zedan. Zedan has always had the ability to see the big picture, and to keep me focused on a goal that could sometimes be quite difficult for me to see. Also, Zedan's open door policy has provided me with endless support and encouragement, which is extremely important for any doctoral student. No matter how busy he has been, he has always found time to answer my frequent questions. I do not think that I could have had a better supervisor for my thesis.

I am extremely grateful to Dr. Aladdin Ayesh, who has been a never-ending source of encouragement and help. He has provided me with many opportunities to develop and apply my research. Ayesh is always enthusiastic and encouraging, and is a continuing source of support for me.

I also would like to thank my all friends and staff at STRL research group especially Mrs. Lindsy, Mrs. Lynn, Dr. Monika Sulti and Dr. Antonio Cau for their help. Being with them made working in the lab very enjoyable. I will always remember our coffee and lunch breaks during which we debated a wide range of different topics, from science to religion, from technology to social problems. These have been a valuable counterweight to my research.

I am extremely grateful to my special friends Mohammed Al-Noim, Raheel Shah, Younis Burhan, Sami Al-Mubarki, Mushrif Al-Ruily, Mohammed Aqeel, Ahmed Al-Hakmi, Khalid Al-Jabran, Ajlan Alajlan, Mohammed Al-Ansari, Abdulwahab Bokhari, Mohammed Mulla and Dr. Ziad Abu Onq, Dr. Abdulmajeed Al-Mubarak and Dr. Abdulwhab Shahin for supporting me and for their encouragement and help. Alnoim and Shah have provided me with the technological support. Especially preprocessing issues have been challenging, and this is an area in which AL-Noim, Shahin, Al-Mubarki and Shah have helped me. I have also enjoyed working with everyone in Text data Mining.

Last but not least, my thanks go to my family, especially my parents, and my brothers and sisters, who have always supported me during my studies, and provided advice and encouragement throughout my research. Finally, I would like to thank my wife, my sons Ahmed and Salman, and my daughters for being there for me and supporting me in my decisions; they truly share in this achievement. Thank you, for your patience, love, and dedication.

Symbols Index

AFP	Agency France Presse
ANN	Artificial Neural Network
AMESD	Arabic Morphology & English Stemmer Dictionary
ASCII	American Standard Code
ASMA	Arabic Stemmer Morphological Analyser
BMU	Best Matching Unit
DSIR	Distributional Semantics based Information Retrieval
HTML	Hyper Text Markup Language
IE	Information Extraction
IR	Information Retrieval
KE	Knowledge Engineering
LDC	Linguistic Data Consortium
MDS	Multi-dimensional Scaling
MMA	Multilingual Morphology Analyser
ML	Machine Learning
MSA	Modern Standard Arabic
MT	Machine Translation
MTM	Multilingual Text Mining
NLP	Natural Language Processing
MLTextMAES	Multilingual Text Mining for Arabic and English Scripts
NN	Neural Network
PSR	Prefixes & Suffixes Removal
QE	Quantization Error
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
IREP	Incremental Reduce Error Pruning

SOM	Self-Organizing Map
SOMMLTMA	Self-Organizing Map for Multilingual Text Mining Algorithm
SOM-Get-Catg	Self-Organizing Map and Get Category
SOM-Get-Dict	Self-Organizing Map and Get from Dictionary
SOM-Get-Docs	Self-Organizing Map and Get Documents
TM	Text Mining
TC	Text Categorization
UTF-8	Unicode Transformation Format 8-bit
VSM	Vector Space Model
XML	Extensible Markup Language

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Objectives

- The motivation behind multilingual text mining.
- To identify the research questions.
- To illustrate the aims and objectives of the thesis and highlight original contribution.

1.1 Background

Information Technology (IT) has created many changes in all our lives. With the development of online technology, such as the Internet, a wide variety of transactions can be made globally.

Technological development has become so visible in all fields that it is now crucial to take it into consideration in every situation. Such development has resulted in the great amount of information that exists nowadays, and one frequently hears the

expression "the information revolution". Today's world is facing the problem of an abundance of information to a point where it is becoming very difficult to utilize the information in a constructive manner.

Text mining, also known as text data mining or knowledge discovery from textual databases has a very high commercial value [3]. Text data mining is the process of discovering useful or interesting patterns or trends from within unstructured text. It is also used to describe the application of data mining techniques to automated discovery of knowledge from text [4]. However, text mining is a more complex task than numerical data mining, which deals with a fix set of features for all analyzed items in a straightforward manner and is very close to machine language (binary)(e.g. age, income, gender, etc.), while the extracted information from text ("string" in computing language), needs to be sorted, compare, manipulate and then also to translate them into machine language. Data mining usually deals with structured data, while text mining is usually fairly unstructured that needs preprocessing stage to clean the text.

Today, the use of methods able to mine sensible linguistic elements from multilingual text collections, with vast amounts of high dimensional data, is a key element in a wide range of applications. In this thesis, we present a new framework for multilingual text mining, based on the SOM for Arabic-English textual data. Through our modified algorithms, we intend to improve upon current multilingual text mining approaches, and to reveal more effectively the patterns and connections that lie within text data. We have used Matlab 5 [5], as the software package used for training the SOM technique.

1.2 Problem Statement

The Arabic language belongs to the Semitic family of languages. The words in such languages may be formed by modifying the root itself internally and not simply by the concatenation of affixes and roots as occurs in an inflecting (such as Latin), agglutinating (such as Turkish and Japanese), or incorporating languages (Greenlandic Eskimo and Chinese) [6, 7]. This type of processing is known as morphology.

Arabic morphology has a great impact on word formation and may appear in a text in different morphological variations. Using morphological analysis to support text mining in Arabic has lead to some different points of view. The root performs well or better than the stem at low recall levels, and definitely better at high recall levels, as Al-kharashi mentioned [8]. Moreover, Abu salem [9] repeated Alkharashi's experiment and found that the root retrieval method performs significantly better than the stem method. Furthermore, Hmeidi [10] confirmed that the performance of the root method is more effective than the words method. He justifies his view by stating that Arabic is a derivative language, and therefore the system should be based on the root of the word. Suppose that the user wants to search for the word (AktetAb,اكتتاب), if the system is based on word search only (See an example Section 4.3.1) then it will retrieve the word as it is entered, but if the user enters an incorrect spelling, then the system cannot retrieve the word. However, if the system is based on root searching, then it will retrieve all forms of the root (ktb, كتب).

Unfortunately, neither group offered clear evidence for their claims. As mentioned above, the support for the root method was only based on giving examples in Chapter 4. As far as academic research is concerned, it is not enough to say that the root method functions better or worse than other methods without any proof to support or reject that belief. However, the work of some other researchers, such as that of Al-kharashi [8], Abu salem [9], and Hmeidi [10], offer a number of experiments investigating the retrieval performance of word, stem, and root methods.

The underlying motivation driving the research is to create a text-mining framework that can extract non-trivial information from an Arabic-English corpus. This research will focus on the text clustering process. One of the key contributions of this thesis is the integration of a new multilingual text clustering method. In the past few years, the Arab world has witnessed a number of attempts to develop Arabic text mining systems, and the current study is one of these attempts. However, since the computer was introduced to the Arabic text and text mining environment, a number of problems have arisen (for example, language issues such as morphology, and processing of very large datasets for text mining). Some of these problems have been solved such as infix and broken plurals, while others remain unsolved as a computational linguistics such as two letters word (nom, نم, kol, كل).

We have placed the focus on text mining bilingual Arabic and English text data, and the reason for this lies in modern history. The countries of the Arabian Gulf have developed enormously since the discovery of oil in the1930s, and this has dramatically affected the lives of the millions of people living there in terms of lifestyle, commerce and even security. The discovery of oil has introduced the English language as a new paradigm, linking it to Arabic in many ways. For example, fiscal reports are written in both languages, even goods labels are written in the same way. Nowadays, the world is suffering from terror as a global challenge. Therefore, reports sent between Arab and non-Arab countries need to be written in both languages, Arabic and English. This thesis focuses on multilingual text mining where the Arabic language is the target language. This, of course, will lead to many challenges and these are discussed in Chapter 2.

1.3 Research Questions

Information on the net is available multi-lingually, but unfortunately the traditional search engines retrieve information from monolingual database in response to the

user query. Although in some instances, results in other languages are produced but most oftenly are not related to the user query.

In this context we propose to address the overall research question that tackles each of the underlying issues.

RQ1. How can we improve the quality of search results looking into pages from different languages?

RQ2. Does the morphological complexity of the Arabic language have adverse effects on mining performance?

1.4 Aim and Objectives

Similar to English, a large amount of unstructured data is available on the net in the Arabic language, but text mining tools and techniques are mostly English language oriented. There is therefore a lack of efficient text mining tools to cover both Arabic and English simultaneously. Thus our aim in this thesis is to provide a multilingual text mining framework to be applicable to Arabic-English bilingual text mining problems. Our approach towards the development of the framework is composed of the four main stages of text mining viz: pre-processing, clustering (SOM), test of quality and graphical user interface. An overview of these four stages of text mining are defined in Figure 1.1 below. In order to achieve the above mentioned aim the following objectives are identified.

1. To develop a novel morphological analysis for Arabic roots.

2. To develop a language-independent approach for the discovery of indirect knowledge from bilingual information sources.

3. To investigate the potential of using SOM in the process of investigating large

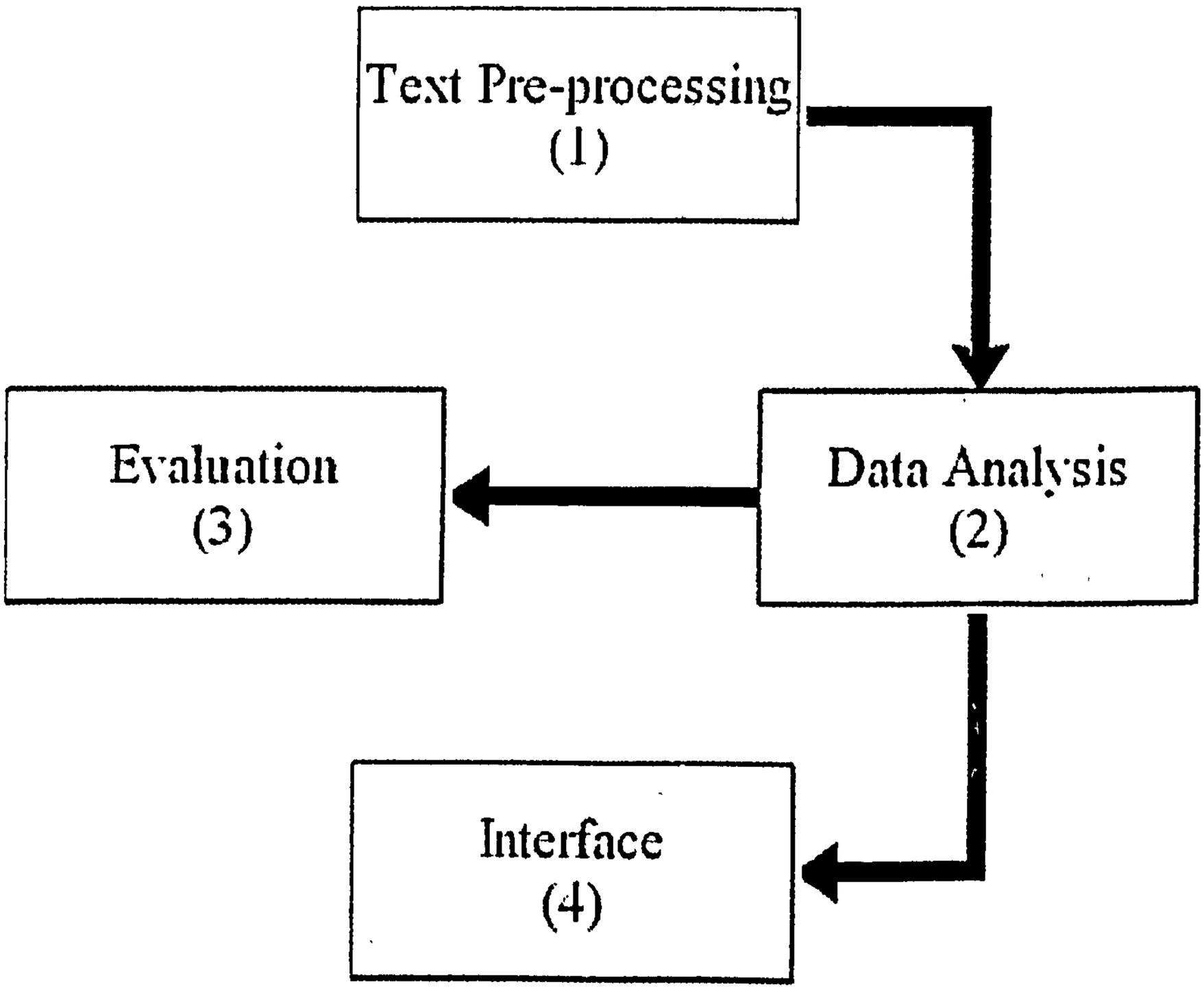


Figure 1.1: Text Mining General Framework

amount of Arabic and English documents. SOM is an efficient tool capable of processing monolingual and multilingual corpora. Since it is based on neural network algorithms, it can be easily implemented in an unsupervised mode to create clusters. It is best suited for visual outcomes as it actually reduces a multi-dimensional input vector to a two-dimensional output.

1.4.1 Contributions of the Thesis

While there is a growing interest in the general topic of text mining, there are few working systems or detailed experimental evaluations. The contributions that this thesis makes to research in the field of multilingual text mining is briefly summarized below:

- Most earlier work on text mining focused on the monolingual. The primary contribution of this thesis introduces MLTextMAES, a new framework for

multilingual text mining based on a SOM for Arabic-English corpora.

- To address the Arabic morphological problem, we developed a code: AraMorph, for stemming the suffixes, prefixes and infixes. With encouraging results from experiments in several domains, we show how these approaches can produce accurate results despite the Arabic morphological complexity.

- Clustering is often performed using the unsupervised learning mechanism. This thesis extends the use of unsupervised clustering technique (SOM) to perform the clustering in specified domains. An algorithm is proposed which combines adaptation and modification to the existing SOM algorithm. The purpose of this work is to investigate the suitability of this technique to classify documents in data text mining, based on the generally accepted guidelines from the literature.

Conclusions concerning technical construction, such as pre-processing issues, are made. The model is evaluated through a number of experiments into the subject matter. The results show that the MLTextMAES model is efficient and fairly accurate when compared with human performance discussed (see Chapter 7). Based on the results achieved, a number of conclusions and recommendations for building a prototype for more extensive testing are presented.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows. In this chapter, an introduction to the thesis has been provided. The aims and the objectives of the research have been defined.

In Chapter 2, the literature in the area of text mining and multilingual text mining is reviewed. The basic components, applications and challenges of text mining are described. A formal definition of text mining and a brief overview of currently

available text mining methods is also presented. This overviews the current state of the discipline.

In Chapter 3, neural networks, especially SOM, are discussed. Firstly, the background and definition of neural networks are given. Then, the SOM is presented in detail, how it works, how we prepare our output to be evaluated by it, and how this system is implemented.

Chapter 4 describes the structure of the Arabic language. This chapter also discusses natural language processing (NLP) in Arabic and English, and it also contains a description of some morphological analysers for both languages and presents a brief review of them.

Chapter 5 presents our methodology for the proposed text mining framework

Chapter 6 describes how text is classified and how the implementation of ML-TextMAES, data generated from our framework is presented. Firstly, properties of the data, and the required pre-processing, are discussed. Then, the training process of the model is explained, and finally the identification of the clusters on the map are illustrated.

Chapter 7 presents the corpus evaluation methodology used to test our system, and results are discussed.

Finally, Chapter 8 concludes the thesis and gives directions for future research.

An overview of the chapters in this thesis is illustrated in Figure 1.2 below.

1.6 Training Package

The Training Package has software specification requirements, and these are specified in detail below. The Operating System employed the following:

- Windows XP Professional Edition.

The Hardware Minimum Specification:

- Pentium III workstation, 500 MHz.
- 512 MBytes of RAM.

The Software required:

- Java Environment.
- Matlab.
- SOM Package.
- The Data.

The Software required can be downloaded from [5], as can the data preparation packages attached to this thesis. A Java package can be downloaded from [11]. Appendix E includes the code of the data preparation packages and training developed for the specifications, and is explained in the MLTextMAES model Chapter 5.

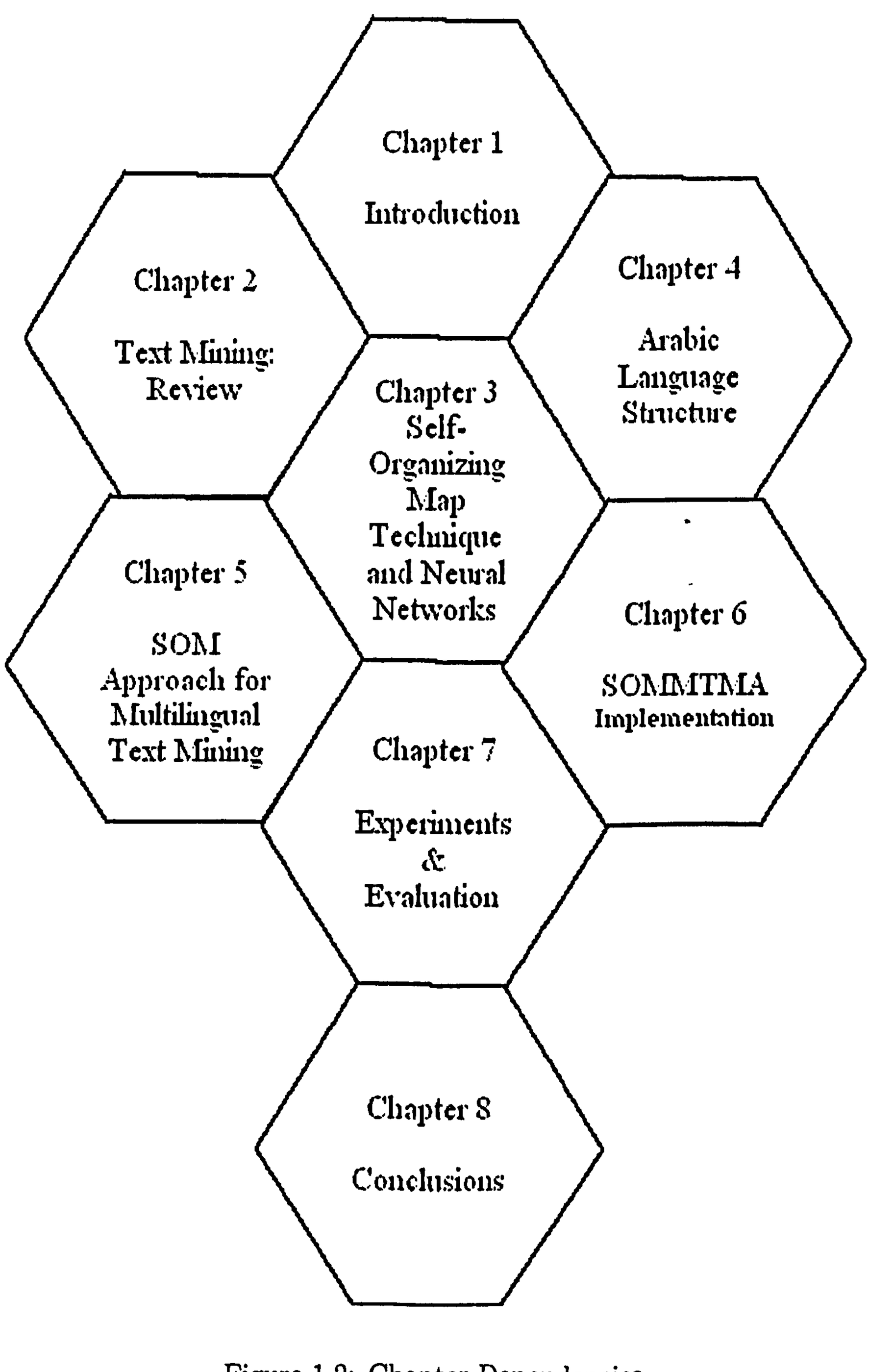


Figure 1.2: Chapter Dependencies

Chapter 2

Text Mining

Objectives

- To introduce text mining and its applications.
- To review formal approaches of multilingual text mining.
- To discuss our observations.

2.1 Introduction

With recent advances in technology, increasing quantities of data are being stored electronically and a great deal of this is in the form of text. Indeed, as much as 80% of a typical company's database is text documents [12]. The possibilities of exploiting such databases for commercial gain is well recognized but the foucs of this research is specifically on the ability to mine the text-based data. Text mining uses modified forms of data mining techniques in order to reveal information inherent within large bodies of text in such a way that it is usable to the researcher. These techniques

are information retrieval, information extraction, data pre-processing, text analysis, text visualization [13] and classification (k-means, Expectation-Maximization), decision tree(C4.5), Boostexter by Schapire, and Rule induction (RIPPER/IREP). More broadly most research employs natural language processing and database technology [14].

Text Mining methods have been widely used in many different areas such as homeland security and intelligence, health care, law enforcement, public safety and bioinfomatics [15, 16].

Text mining tools focused primely on processing monolingual documents (particularly English documents) but researchers have paid little attention to applying the techniques for handling the documents in multilingual information sources [17].

We will give an overview of the ways in which these methods are applied to text mining. We focus on these techniques because in our current research we primarily rely on these fields for text mining.

2.2 Definitions and Rationale

Several different operational definitions of text and data mining have been proposed and we list some of them below:

Mladenic and Grobelnik [18] have defined text mining as follows: *"The objective of Text Mining is to exploit information contained in textual documents in various ways, including discovery of patterns and trends in data, associations among entities, predictive rules, etc."*

Tan [12] has defined text mining as: *" Refers generally to the process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents. It can be viewed as an extension of data mining or knowledge discovery from (structured) databases".*

Dzeroski and Dzeroski and Wrobel [19] have defined data mining as follows: *" is the multi-disciplinary field dealing with knowledge discovery from relational databases consisting of multiple tables".*

Thabtah and Cowling and Peng [20] have defined data mining as: *" The process of discovering hidden information from data sets for the purpose of prediction.".*

The core difference between data and text mining is the retrieval of information from structured and unstructured databases, respectively.

We will give a different definition of text mining, which is motivated by the specific perspective of the area.

Definition 2.1: *Text mining may be seen as operation that identifies useful knowledge and extracts non-trivial patterns from unstructured documents.*

Text mining is extremely useful since it enables us to analyse and classify large amounts of textual data and to reveal the knowledge buried in it. Neri [21], asserted that text mining can be used as an intelligent and powerful search engine. Below are some points showing how important text mining is, and how it can help business [22].

- It allows users to access documents by their topics.
- It transforms huge volumes of data into detailed information, providing an overview of its contents.
- It helps users to discover either hidden and meaningful similarities among documents or any related information.
- It explores how a market is evolving.
- It looks for new ideas or relations in topics.
- It identifies and solves problems

2.3 Text Mining background and Development

Text mining is extraction of a non-trivial information that is relevant to the researcher. It gives an enquirer the ability to seek only the information he/she needs from within a mass of largely irrelevant information. Recently an Arabic-English corpora has emerged and it is the focus of this research to enable enquirers to mine information from such multilingual information sources.

It is entirely possible that large bodies of text-based information contain knowledge that is unknown or hidden, and [23, 24] view text mining as the opportunity to bring to light knowledge that hitherto would not have been possible. However, text mining is problematic as computational linguistics is a complicated field, and multilingual computational linguistics even more so. These matters are addressed by Hearst [3] in her work on the problems in data mining, information access, computational linguistics and text data mining.

Nevertheless, steps are being taken forward and, [25] in his work on Arabic language text mining described new developments in increasing the rate at which information can be extracted through roots-based hierarchical indexing method. He was even able to demonstrate that gains in increased computational speeds of the range 50%-100% can be achieved for typical queries.

Hitherto, most the research into text mining technique has been into databases that are monolingual, and the majority of this has been alone on English language texts. However it would be revealing and profitable if one could mine multilingual texts, and it is the purpose of this thesis to include contributions such as the stemming, morphological analyser and modify monolingual text mining techniques for mining multilingual datasets. This shall be done with SOM and will concentrate on Arabic-English datasets.

This work is related to Knowledge Discovery in Databases (KDD), and [26] identified this as *"the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data"*.However, this is largely concerned with numeric data and text mining seeks meaningful patterns within unstructured textual data [27].

2.3.1 Machine Learning

An excellent survey has been given by Sebastiani on machine learning (ML) and text categorization (TC) [28]. According to him the mathematical definition of TC is that:

"Text categorization is the task of assigning a Boolean value to each pair $(d_j , c_i) \in D$ *X C, where D is a domain of documents and* $C=\{c_1, ..., c_{|c|}\}$ *is a set of predefined categories. A value of T assigned to* (d_j , c_i) *indicates a decision to file* d_j *under* c_i*, while a value of F indicates a decision not to file* d_j *under* c_i*".*

The important applications of TC are given. For example, document organization which is indexing with a controlled vocabulary is an instance of the general problem of document base organization. For instance:

"at the offices of a newspaper incoming 'classified' ads must be, prior to publication, categorized under categories such as Personals, Cars for Sale, Real Estate, etc. Newspapers dealing with a high volume of classified ads would benefit from an automatic system that chooses the most suitable category for a given ad. Other possible applications are the organization of patents into categories for making their search easier".

The author has emphasis on proper usage of ML for:

1. Text Filtering.

   The classification of documents, especially when it happens between a data provider and his client, is called 'text filtering'. Such filtering usually occurs between a news agency (provider) and a newspaper (client), where the filtering is designed to only allow the delivery of news that the client probably wants, for example filtering in order to only deliver sports news to a sports paper. Text filtering classifies documents into two separate categories, one which is of use to the client and another which is not. It is therefore a useful manifestation of single-label text categorization.

2. Word Sense Disambiguation.

   Identifying correct word meanings can be fraught with difficulty, as polysemes and homonyms exist in every language. Word sense disambiguation is the task of overcoming this, for example determining the correct meaning within a given context of words such as 'invalid' (noun or complement) or 'get' (buy or receive). For instance, Everest may have (at least) two different senses in English, as in the Everest related to mountain or Everest of manufacturer.

3. Hierarchical Categorization of Web Pages.

   Researchers are currently examining the possibilities of exploiting Text Categorization for improving the performance of Web search engines. Hierarchical catalogues could be constructed so that Web pages, or sites, could be automatically classified by an Internet portal. Thus when a query was entered into a search engine, the algorithm would firstly explore a more limited number of categories before seeking the document desired by the enquirer.

Detailed section is devoted to the main ideas under highlighting the ML approach and TC. Initially, he compares the Knowledge Engineering (KE) techniques with

Machine Learning technique (ML), and he outweighs the KE by providing evidence in favor of ML, extract is given bellow:

1. Knowledge acquisition well known as bottleneck from the expert systems. That is, the rules must be manually defined by a knowledge engineer with the aid of a domain expert (in this case, an expert in the membership of documents in the chosen set of categories): if the set of categories is modified, then these two professionals must intervene again, and if the classifier is ported to a completely different domain (i.e., set of categories), the various domain expert requests to intervene and the job has to start from the beginning.

2. The advantages of the ML approach over the KE approach are clear. The engineering effort goes in the direction of the construction not of a classifier, but of an automatic builder of classifiers (the learner). This means that if a learner is (as it often is) available off-the-shelf, all that is required is the effective, automatic construction of a classifier from a set of manually classified documents. The same happens if a classifier already exists and the original set of categories is updated, or if the classifier is ported to a wholly various domain.

Approaches to TC is dealt in detail various indexing schemes are discussed, for example document indexing and the darmstadt indexing approach, another aspect of TC is the dimensionality reduction, two approaches of dimensionality reduction are explained that is term selection and term extraction. Also, the author tackles in detail inductive construction of text classifiers from a "training" set of pre-cassied documents. Text calssifiers are also evaluated of the basis of different statistical parameters, for example, (*precision and recall, recall, accuracy, efficiency, etc.*)

The survey supports our decision to tackle issue of Arabic-English documents clustering by using machine learning technique (SOM) and filtering approaches discussed in the section of pre-processing of our general framework in Chapter 5.

2.3.2 Information Retrieval (IR)

Information retrieval is not only concerned with retrieving but also with organizing, storing and searching of information from within datasets. This is greatly aided by the employment of key words; which are found in the document and which can be used to represent it. This concept was first introduced by [29] and they called this idea is the Vector Space Model (VSM). In this model, any number of key words are grouped into a "vector" which then represents the document. The process of VSM model has three stages: document indexing, term weighting, and computation of similarity coefficients.

Stage 1, in document indexing, each document is represented as a vector in a high-dimensional space. Stop-words in a document do not describe the content, and included "an", "and" or "the". By removing non-significant (common) stop-words from the document vector, the vector represents only meaningful words. The size of the vector should be less than 40-50 percent of the total number of words in the document [29].

Stage 2, the weight of the indexed terms supports the retrieval of information which is relevant to the user query. Terms are weighted to indicate their importance for document representation. Most of the weighting using inverse document frequency or generated randomly.

Stage 3, ranks the document with respect to the query according to a similarity measure. The similarity between any two documents reflects the a distance between documents in a high dimensional input space. The most popular similarity model for estimating the similarity between vectors uses correlation co-efficients (Cosine of the angle between vectors) [30].

2.3.3 Boolean Model for IR

A query may be entered using Boolean parameters, and such a query is formulated with the classical operators AND, OR, and NOT. The query "q1 AND q2" is satisfied by a given document D1 if and only if D1 contains both q1 and q2. Similarly, the query "q1 OR q2" is satisfied by D1 if and only if it contains q1 or q2 or both. The query "q1 AND NOT q2" satisfies D1 if and only if it contains q1 and does not contain q2. More complex Boolean queries can be built up out of these operators and evaluated according to the classical rules of Boolean algebra, and result in a query being either true or false i.e. a document either satisfies such a query (relevant) or does not satisfy it (non-relevant).

The classical Boolean model does not use term (query) weights as such weights should be restricted to between 0 to 1. In a Boolean model, each coordinate of a document vector is zero (when the corresponding attribute is absent) or unity (when the corresponding attribute is present). There are many adaptations of the Boolean model, and the most common are term-weighting is a widely used refinement of Boolean models that takes into account the frequency of appearance of an attribute (such as keywords and key phrases) or location of appearance (e.g., keyword in the title, section header, abstract or texts) [31, 32].

2.3.4 Probabilistic Model

In IR it is often not possible to differentiate between probabilistic and statistical methods. Indeed, there are many similarities as probabilities are often calculated on the basis of statistical data. The literature shows that probabilistic IR results have most employed statistical data. Naturally, where various types of data are being used in a probabilistic model [33], the data is often collated and presented to the formula in a statistical manner. Because of the similarities between these models, the same data may be utilized, e.g., tf's and idf's weight schemes, or the

tf-idf (the probability that combines the 'within document-term' frequency with the inverse within the data) [34].

2.3.5 The TF-IDF Weight

The term frequencyinverse document frequency (TF-IDF) weighting scheme is used to assign higher weights to distinguish terms in documents [29]. TF-IDF makes two assumptions about the importance of a term. First, the more a term appears in the document, the important it is (term frequency). Second, the more it appears through the entire collection of documents, the less important it is since it does not characterize the particular document well (inverse document frequency). Actually, TF-IDF works by determining the relative frequency of terms in a specific document compared to the inverse proportion of the total number of documents to which each term is assigned. This weighting scheme reflects the importance of a term occurs within the document itself and the corpus as a whole. We see that the importance of a term is not linearly proportional to the IDF. Indeed, terms that occur extremely rarely have very high IDF while common terms thus receive a very low TFIDF [29, 35, 36]. The framework procedure of the inverse document frequency can work as follows. The TF-IDF is given by:

$$tf - idf(i) = tf.log(\frac{Ndoc}{df(i)}$$

where i term,

- Ndoc is the number of all documents in the collection.

- tf(i) is the term frequency (number of word occurrences in a document).

- df(i) is the document frequency (number of documents containing the word).

- tf-idf(i) is relative importance of the word in the document.

2.3.6 Information Extraction (IE)

From a large unstructured database, a researcher seeks to take out specific information for analysis. This process is called information extraction and it stores this information as patterns. This process has two tasks: identifying the specific field to be extracted e.g. name, date, or address, and using natural language processing (NLP) technology to identify speech with the text [37].

Let us examine the following text to show that the main task is to extract parts of the text and identify specific attributes to it, for example to extract executive position changes from a news story: "Robert L. James, chairman and chief executive officer of McCann-Erickson, is going to retire on July 1st. He will be replaced by John J. Donner, Jr., the agency's chief operating officer".

In this case we may identify the following information to be extracted:

Organization name :	McCann-Erickson
Position :	chief executive officer
Date of change :	July 1*st*
Person leaving :	Robert L. James
Person replacing :	John J. Donner Jr

2.3.7 Data Preprocessing

In order to ease the process of retrieving and analysing data, any database under consideration should be cleaned. This is called data pre-processing and involves the use of an algorithm that removes stop-words, common words, conjunction, pronoun, punctuations marks and html tags from the documents. This procedure also generates the key words. The purpose of the above is to produce data that is comprised of word roots, i.e. text, that has no prefixes or suffixes. This is achieved through a "stemming" algorithm, and can produce sets of stem classifications. In Arabic, the

most fruitful methods have been "light" and "root-based" stemming algorithms, as reported by [38] in his overview of the subject.

2.3.8 Text Analysis

The above processes, including the creation of vectors, takes place in high-dimensional input space, but analysis requires a reduction in this dimensionality into only a two-dimensional output space. Such reduction necessitates the employment of cluster analysis.

Cluster analysis is the process of grouping together items from within an data set that are similar in some way. The parameters for this are determined by the researcher but at the end the data is divided into a number of clusters. Each cluster is dissimilar in some specified way but contains data items (e.g., documents or terms) that are similar to each other [13]. Lately, the advent of World Wide Web search engines, and the concept of text data mining has led to a renewed interest in clustering analysis. Effective clustering can be of great benefit to disciplines which handle large quantities of data. In the followings fields clustering techniques are highly beneficial:

- Gene identification / analysis: finding similar DNA over several condition or protein sequences [39].

- Disease diagnoses: finding a virus similar to a given one from a large virus dataset, or finding groups of viruses with certain common characteristics[40].

- Social services: clustering to identify groups with particular requirements (e.g. the elderly) would economize on resources, and improve its allocation [41].

- Document retrieval: mining documents related to a given query into a map [42].

- City planning: identifying groups of houses and buildings according to type, use, architecture, value and location [43].

- World Wide Web: clustering or finding sets of related pages [4].

- Marketing: helping market analysts identify distinct groups in their customer base, so that they can exploit this knowledge in the development of marketing strategies [43].

- Predication: is different from classification and estimation in that the objects are classified according to some predicted future behaviour or to some estimated future value. The SOM technique is applied to a data set and the results are clustered, based on the features of the objects that belong to these clusters. Accordingly, unknown objects can be classified into specified clusters based on their similarity to the clusters, and so useful knowledge related to the query can be extracted. For example, a cluster process is applied to a dataset concerning patients infected with the same disease. The result shows the number of clusters of patients, according to their reaction to drugs. Then, for any new patient, we can identify the cluster into which he can be classified, and thus determine any medication decisions that could be made [44, 45, 46].

2.3.9 Text Visualization

A general goal of analysis is to detect meaningful underlying dimensions that allow the researcher to explain observed similarities or dissimilarities (distances) among the investigated objects. This is accomplished by solving a minimization problem such that the distances among points in the conceptual low-dimensional space match the given (dis)similarities as closely as possible. In factor analysis, the similarities among objects (e.g., terms) are expressed in the correlation matrix. With Multidimensional Scaling (MDS), one may analyze any kind of similarity or dissimilarity

matrix, in addition to correlation matrices. However, a major weakness of MDS is that there are no quick and fast rules to interpret the nature of the resulting dimensions.

2.3.10 Self-Organizing Map (SOM)

There is a number of clustering techniques, but this thesis will provide description for the SOM algorithm that is most widely used in document clustering. The SOM algorithm is based on the idea that a winner, which after reducing the high-dimensional input space into low-dimensional output grid, can represent a cluster. The basic SOM algorithm for finding a winner cluster is presented below:

1. Set initial parameters.
2. Compute the distances of a winner.
3. Assign the neighbourhood to the closes winner.
4. Update the weight matrix of the winner and the neighbourhood.
5. Repeat steps 3 and 4 until the weights do not change.

One of the best-suited methods for text-mined data visualization is SOM. It employs artificial neural networks that work with unsupervised learning [47, 48]. This important technique was invented by Kohonen to aid in the visualization of information [49, 50]. SOM has two particular uses; as a statistical tool for multivariate analysis because it has the ability to reduce multi-dimensional data document into two-dimensions output space, and as a clustering tool because it very effectively group similar data points together. During the learning process, the algorithm used in SOM alters the weight vectors of each data item and presents them on a two-dimensional grid map. This facility is the reason that SOM is widely used for mining

text and for visualizing large dataset. Application fields include, image processing and speech recognition, process control, economic analysis, and diagnostics in industry and in medicine. Details of SOM will be discussed in Chapter 3.

Many other clustering techniques have been developed, which work well in particular scenarios. The most commonly used algorithms are K-means, Expectation-Maximization, Decision tree (C4,5), Bosstexter and Rule induction (RIPPER/IREP). Below is an overview of these algorithms these algorithms where we explain why they may or may not be relevant to text mining. We shall not focus heavily on these algorithms because we are primarily concerned with text mining in our current research.

2.3.11 K-Means Algorithm

K-means algorithm developed in 1967 by MacQueen. It is an extremely popular partttional clustering Technique, and it has been utilized in a great many different disciplines; its clustering algorithms can be used in, for example, data mining and compression, probability density estimation. Nevertheless, it has a drawback in that the user must determine the value of k (number of clusters) for the k-means clustering algorithm. An optimal value for k is not always apparent, and selecting k through an automatic process is computationally difficult. High dimensionality makes the selection of the value of k even more problematic, even when the desired clusters are identifiable.

Furthermore, the iterative process of this algorithm often terminates prematurely, and therefore a partial albeit efficacious result is sometimes realized. An additional problem lies in its random selection of initial centres, which often provides unstable results. This is because clustering is often applied on data where the end user is unable to judge the clustering quality. The literature shows that many researchers have attempted to identify the key features of k-means clustering and any characteristics

that impact on the clustering analysis, and they have identified high dimensionality, the size of the data, the sparseness of the data, noise and outliers in the data, types of attributes and data sets, and scales of attributes [51, 52, 53]. This algorithm follows the four steps below:

1. Select random initial k points into the space represented by the objects that are being clustered.

2. Calculate the distance between each object and the group that has the closest centroid.

3. When all objects have been assigned, recalculate the positions of the k centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move.

2.3.12 Expectation-Maximization Algorithm (EM)

EM is an iterative procedure that efficiently locates maximum likelihood estimates, even where there is missing or hidden data. The EM algorithm uses a probability distribution estimation in order to select the number of clusters, and the creates mixture models that describe the results through statistical distributions. This procedure is not unlike that of k-means as a set of pre-determined parameters are subjected to iterations until a desired clustering or convergence is obtained. The iterative procedure of the EM algorithm has two parts: the E-step, and the M-step. The first is the expectation step, where the observed data (or estimates of unknown parameters) are used to make estimations for any missing data. This step is performed through conditional expectation, hence E-step. In the second step, the likelihood function is maximized, hence M-step. This assumes that the missing data are known by substituting the missing values with the estimated missing ones that

were determined in the E-step. The algorithm is specifically designed to increase the likelihood of convergence at each iteration [54, 55, 56]; its steps are as follows:

1. Select an initial set of model parameters (randomly)
2. Repeat.
3. Expectation Step: For each object, calculate the probability that each object belongs to each distribution.
4. Maximization Step: Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
5. Continue until the parameters do not change (or change below some threshold).

However, in the practical example of the dialysis patient dataset, the distributions for the features are unknown, and therefore a computation of the probability for each data point is not possible.

2.3.13 Decision Tree (C4.5) Algorithm

In order to extract meaningful knowledge, a database must be subjected to classification. Decision Trees use efficient, relatively easily understandable algorithms, and are therefore popular techniques for prediction in machine learning and data mining. Additionally, the models are easy to interpret so the C4.5 classification algorithm has been studied and employed a great deal. It builds decision trees by dividing and sub-dividing the data into any number of predetermined classes, and does so heuristically [57, 58].

2.3.14 BoosTexter Algorithm

Actually employs two slightly different algorithms together with a one-level decision tree for base classification. The tree is a weak learner and is sometimes called a

decision stamp. The purpose of the first algorithm, called ADABOOST.MH (or ADABOOST in Schapire) is to maximize the effectiveness of the micro-averaged data. The other, called ADABOOST.MR, is to minimize any ranking loss by assigning correct category rankings to every document in the dataset [28, 59].

2.3.15 Rule Induction (RIPPER/IREP) Algorithm

[60, 61] is a popular system where sets of rules are involved in learning. Rule sets are relatively easy to understand and use, and this system is often more effective than a decision tree. Additionally, some prior knowledge of the data can be included in the rule learning stage. The rule sets employ a first order system (prolog predicates), which researchers find both natural and familiar, as the learning of propositional rule sets can usually be widened to incorporate a first order system. Unfortunately, rule learning systems do not usually scale particularly well in accordance with sample sizes, and this is generally the case with noisy data. Two of the more commonly employed schemes for rule learning are C4.5 and RIPPER. The former is addressed above. The latter is based on Incremental Reduce Error Pruning (IREP), and is a fast rule learner, which divides the available training data into two sets (one for growing and one for pruning) before proceeding to the learning phase. Thus IREP can work with set-valued features, and has a 2-class approach that divides a data set into two subsets. The first one (positive set) constitutes example features of the desired class, whilst the other (negative set) is made up of samples of the rest of the data.

2.3.16 Text Mining Applications

Organizations can mine unstructured text from anywhere - from sources such as file servers, or from any place that keeps free text data.

Common application areas of text mining are follows [62]:

- Marketing: "text mining can be utilize as a state or property agent", ironically speaking if a customer wishes to buy a certain land or house, he/she should lay out preferences and then text mining works as an exact search engine that produces results that correspond only to the customer's preferences.

- Homeland Security and Intelligence, to put terrorist networks under deep investigation in order to identify potential crimes and to detect their roots, hoping to prevent them.

- Law Enforcement, to aid governmental agencies to enforce law and order, by identifying previously unnoticed trends, links and patterns.

- Healthcare Scheme, the opinion of the patient is highly important, and to take this into account we need to analyse patients records and reports to obtain accurate facts and pertinent responses in order to enact the directives that have been laid down by the government.

- Bio-informatics, to obtain the best understanding through mining scientific journals for critical information associated with genes and proteins; e.g., genes and their associated functions, diseases, and tissues.

- Public and Insurance Safety; in order to develop and improve the performance of private and public departments incidents reports should be revised carefully in order to identify rout causes to prevent future errors.

For more details refer to [16]. Specifically there are government research programs, such as Terrorism Information Awareness (TIA - originally Total Information Awareness) proposed by the US Department of Defense, which aim to analyse huge volumes of structured and unstructured data in order to detect patterns and predict and prevent terrorist attacks.

Because of political issues such programs are likely to be funded secretly by governments. Text mining is already used by many companies and government agencies for specialized applications, and this use will increase over the next few years due to security concerns, as will the scope of text mining applications.

An example of exploitation of text mining is "Homeland Security and Intelligence" in the United States. The U.S. Senate Committee on Intelligence [63] showed that before September 11, 2001, the various intelligence agencies had collected a significant amount of data about the individuals who went on to attack the World Trade Center and the Pentagon, but they had been unable to connect the information in a meaningful way. Now text mining tools are being used by the Defense Intelligence Agency, the Department of Homeland Security and FBI to analyse different kinds of intelligence data such as e-mail messages, phone call transcripts, memos and foreign news stories.

Text mining-based programs can work through thousands of documents using statistical and mathematical analysis, to determine how words relate to each other. Text mining has become important in the detection and prevention of crimes on the Internet such as chat rooms. For example, an incident occurred in June of 2000 John and Julie Smith (fake names were given to protect their privacy as witnesses) contacted Child Net to ask for help and advice in dealing with a personal family tragedy: their 13 year old child had been contacted and sexually abused by an adult who met her through a teenage Internet chat room, they asked Child Net for help in alerting other parents to the dangers of chat rooms and ensure that those companies who run chat rooms provided clearer safety advice for children. The team worked closely with the family and child-welfare experts in developing the central aim of the site which is " to raise awareness among children and parents about the potential dangers of un-moderated Internet chat rooms, and to seek to put pressure on those companies providing chat to do more to protect children". For a successful text

mining system the following are needed [62]:

- A well-defined limited query- user query (text based)
- A source of textual documents
- A text mining algorithm
- Pre-processing phase
- A creation component to create new data resource
- A choice of a good area (bio-information: genes, proteins, etc.) to produce catalogues

2.3.17 General Framework of Text Mining

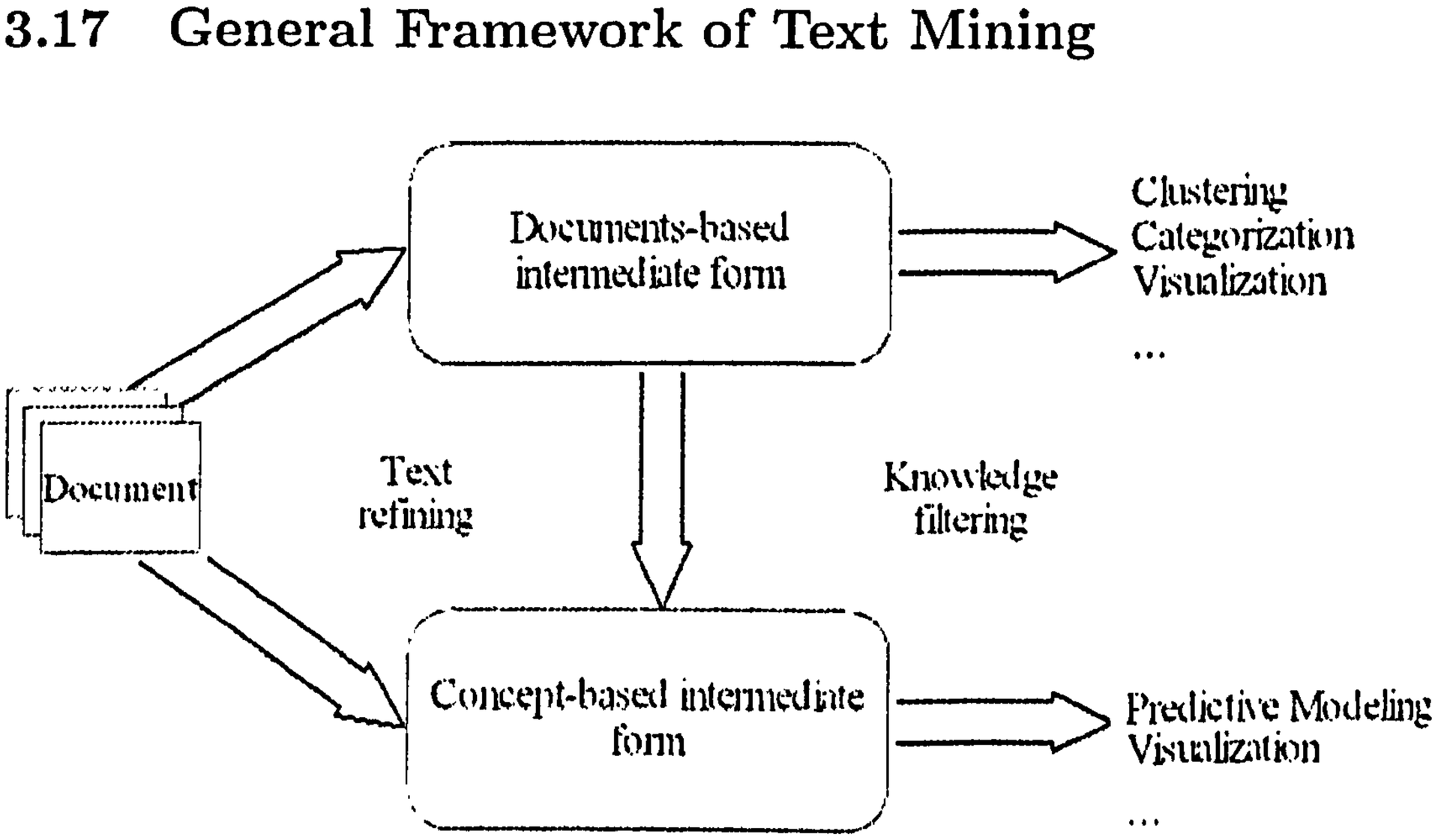


Figure 2.1: Text Mining General Framework

Figure 2.1 which is adapted from Tan [12], shows text mining to consist of two major phases:

1. Text refining
2. Knowledge Filtering

The general framework of text mining as illustrated by Figure 2.1, shows that in the initial phase text documents are classified into document or concept-based intermediate forms (IF). By document-based we mean that each entity is presented as a document, whereas in concept-based each entity represents an object or "concept" of interests in a specific domain. A document-based IF can easily be projected onto a concept-based one. In the next stage, patterns of knowledge can be deduced across documents if applied to document-based IF, and deduced across objects or concepts if applied to concept-based IF. Although this model looks at the text mining of monolingual situations, it can also be applied to bilingual text mining problems. Therefore, we are going to build upon previous work in order to improve the effectiveness of mining Arabic-English documents, and this is reviewed in Chapter 5.

2.3.18 Text Mining Components

From the author's perspective, there are three major components of text mining, which are as follow:[64].

1. Information Retrieval

2. Information Processing

3. Information Integration

Information retrieval is the selection of relevant information from within an unstructured source, or of text segments from a data source, for further processing. Let us examine the following example [65]:

Database type:	Unstructured
Search mode:	Goal-drive
Atomic entity :	Document
Information needed:	"Halal restaurant in Birmingham that serves vegetarian food"
Query :	"Find halal restaurant in Birmingham" or "Birmingham– Restaurant – Halal"

Information processing is the application of bibliometric and computational linguistics and clustering techniques to the retrieved text to typically provide ordering, classification, and quantification for the formerly unstructured material. Information integration combines the computer output with human cognitive processes to produce a greater understanding of the technical areas of interest.

2.3.19 Challenges of Text Mining

Some of the challenges that face text mining are addressed by [65] and are listed below:

1. The problem of data collection from free text:

    - Natural language text contains ambiguities on many levels
        - the use of pronouns (he, she..)
        - the use of synonyms (escape, flee..)
        - the use of words with multiple meanings(Everest related to mountain or manufacturer)
    - The data is not well-organized:
        - the language used in email, chat rooms. e.g. "r u well?", "plz c me!"
    - Semi-structured (e.g. textbook contains table, graph, image, etc.)

- Unstructured (e.g. world wide web).

2. Very high number of possible dimensions, all possible word and phrase types in the language

3. Complex and subtle relationships between concepts in text:

   - e.g."AOL merges with Time-Warner",
   - e.g."Time-Warner is bought by AOL"

4. Ambiguity and context sensitivity:

   - e.g. Automobile = car = vehicle = Toyota
   - e.g. Apple (the company) or apple (the fruit)

5. Unstructured data mining:

   - documents are not structurally identical.
   - documents are not statistically independent.

6. Finding efficient text mining techniques for multilingual corpus.

2.4 Multilingual Text Mining Background and Development

The text mining of multilingual documents has not been the subject of very much research, and therefore the considerable power of computational analysis has not been fully exploited in this field. As a consequence, problems that have been encountered have been left unresolved [3, 17, 66]. Researchers in the past have paid a great data of attention to data sets that are "clean", that is of sufficiently high quality to facilitate scaling. Thus databases that are both large and contain many

dimensions have been analysed or clustered, improving knowledge in many fields. However, multilingual texts have not been so analysed and therefore problems have been created in this field [26].

The purpose of text mining is to reveal precisely the information that is being sought from within a large unstructured database according to a user query [67]. In our research we are building a tool that seeks out connections between documents in both Arabic and English. Our objective is to cluster such information and therefore it is important that we are able to reduce the dimensionality of the vector space. The most faithful method in this is to use artificial neural networks, and therefore to use SOM as one of the major unsupervised learning [68].

2.4.1 Definition of Multilingual Text Mining (MTM)

Although we do not have a common definition for MTM. a general description of its function given by Multilingual Information Discovery is defined as follows:

Search keywords are entered in English and translated into the foreign language (e.g. Arabic). Then the system retrieves relevant documents across multiple languages, which are then presented as results to the user.

2.4.2 Related work

There has been much research in the literature addressing the problem of mining in English. In addition to the English language there has been research into European languages such as France and in Asian languages such as Chinese and Japanese. However, in Arabic language there has been no research using multilingual texts, but there have been some research using monolingual text mining. For multilingual text mining, for instance, the following are of particular interest:

Lee and Young [17] worked on multilingual text-based data and were able to extract indirected information. They developed a technique that was independent of language to do so, but needed a natural-language technique in order to overcome problems in linguistics. Many researchers of text mining techniques have been focused on monolingual text documents, particularly English text collections, and then they have applied those techniques to address documents in other languages such as Chinese. However, the Chinese language is unlike English in many ways and this effects the text mining techniques. For example, words in Chines usually consist of one, two, three, or four letters, sometimes more, but there are no spaces between words. The literature review above has shown the lack techniques available for handle the multilingual text databases, although a few researchers have expanded the text mining algorithm to support various aspects of multilingual text datasets. In light of this, Lee and Young developed a technique in order to address language difficulties when uncovering indirect knowledge from a multilingual text dataset. They conducted several experiments based on a Chinese-English corpus, and some interesting results emerged from those experiments. The SOM was employed to reduce the high-dimensional input space of their textual data into two-dimensional output space by clustering the similar input data close to each other on the map [50, 68]. They used an SOM on a Chinese-English parallel text database and their work has been very useful in our research.

Besancon and Rajman [69] applied the concept of vectors in their multilingual framework in order to identify semantic similarity between texts. They employed a standard Vector Space model and more advance Distributional Semantics based Information Retrieval (DSIR) model and were able to show that vectors can be used for semantics across languages.

Gaussier [66] also worked across languages and was able to mine multilingual bodies of text including clustering, categorizing and retrieving information. However the

success of his results seems to depend on the languages being analysed. The results are not particularly satisfactory.

An adapted version of the M-LaSIE-II IE system was employed by Azzam, Humphreys, Gaizauskas and Wilks [70]. They addressed many aspects of multilingual information extraction but only analysed a relatively small parallel corpus of English-French newswire texts.

Chua and Yeh [71] explored concept-based approach to mining multilingual corpora and were able to show that concepts in the desired information and in the needed information can indeed be matched. They also demonstrated this across languages and showed how a multi-agent system could aid concept-based cross-lingual text retrieval (CCTR).

Maa, Kanzakib, Zhangb, Muratab and Isahara [72] used visible SOM monolingual semantic maps in Chinese and Japanese, clustering words that were semantically similar. They successfully used SOMs and their output maps had clusters whose relative separating distances was a measure of their semantic similarity. They analysed their Chinese and Japanese datasets, clustering co-occurring words that had grammatical similarities. The parameters set for defining those similarities were based on their own word-similarity computation but their results were evaluated numerically. They also used in their evaluations, recall, and the F-measure, and their own human subjective judgements.

Neri and Raffaelli [73] have used the approach used by SYNTHEMA for Multilingual Text Mining, showing the classification results on around 600 breaking news written in English, Italian and French. Typically used three steps process; (the linguistic pre-processing extracts bilingual lexicons from comparable and parallel corpora, enriching existing bilingual dictionaries and helping overcome the language barrier for cross-language information classification) with a text mining. The Classification of documents uses Unsupervised and Hierarchical Clustering.

Tao and Zhai [74] have exploited frequency correlations of words in different languages in the comparable corpora and discover mappings between words in different languages. Also proposed combining four different methods for computing cross-lingual document similarity, including a baseline expected correlation method, an IDF-weighted correlation method, a TF-IDF method, and a translation model method.

Denoyer et al. [75] developed a new model which allows to take simultaneously into account the structure and the content information of electronic documents. This model offers a natural framework for the integration of different information sources. It is based on a statistical framework: Bayesian networks are used to model the documents and to combine the information present in the doxels.

Montalvo et al. [76] have developed multilingual document clustering systems related documents solution. The model can be classified in two main groups: the ones which use translation technologies, and the ones that transform the document into a language-independent representation. One of the crucial issues regarding the methods based on document or features translation is the correctness of the proper translation. Although word-sense disambiguation methods can be applied, these are not free of errors. On the other hand, methods based on language-independent representation.

Table 2.1: Comparative Table

Techniques	[17]	[69]	[66]	[70]	[71]	[72]
SOM	√	×	√	×	×	√
Text Mining	√	×	×	×	√	×
Vector Space	√	√	×	×	×	×
Information Retrieval	×	√	√	×	×	×
Information Extraction	×	×	×	√	×	×
Semantic Space	×	×	×	×	√	√
Translation	√	×	√	√	√	√
Encoding	×	×	×	×	×	√
Frequency	√	×	×	×	×	√
Root-Based	×	×	×	×	×	×
Languages(Arabic/English)	×	×	×	×	×	×

Table 2.1 above shows the different techniques used by various authors. Lee and Young go into great detail in the importance of identifying patterns within multilingual text-based data, and they were able to extract indirected information from unstructured dataset. This influences the proposed work in terms of how the results should be summarized. Additionally, they showed how the capturing of metrics is important in order to allow one to draw meaningful conclusions, and this too influences our proposed work (see Chapter 5).

This research will provide bilingual root dictionary in order to help other researchers determine the most appropriate scripting language solution for a stated problem. The focus of this research extends the work reviewed in this chapter by investigating language difficulties as they apply to text mining. The combination as used by Lee and Young is the best because they have combined the powers of SOM, text mining and vector space [17, 71, 72, 15, 77, 69, 27]. The successful application of this combination to an Chinese-English corpus encourages us to apply the same combination to Arabic-English corpus, and thereby to validate this approach. To evaluate the general framework it was applied to a small set of Arabic-English corpus, which

were then combining automatically into a single lexical knowledge base for use in the first stage of the framework.

2.5 Summary

In this chapter, we have summarized various approaches in text mining system that have been employed by other researchers in this field. The problem is the processing of the very large datasets that are normal for text mining systems, and we have shown how others have addressed the problem of mining enormous qualities of textual data. An essential requirement for these data models is the reduction of very high-dimensional data into two-dimensional data, without the loose of important data, but reducing data noise. There is still much research to be done to satisfy those requirements and to fully establish the suitability of SOM. This chapter also identified those researchers who have mined mono-, bi-, and multilingual databases.

We identified the various stages of text mining, text categorization, information retrieval, Boolean model, Probabilistic model, term weighting schemes, information extraction, pre-processing (cleaning), text analysis, clustering, classification (k-means, Expectation-Maximization), decision tree (C4.5), Boostexter by Schapire, rule induction (RIPPER/IREP), and text visualization. Some of these algorithms are closely related to data mining rather than related to text mining but there are some important differences, for example, the reduction of dimensionality, the use of vectors, the importance of clustering, and overcoming difficulties in linguistics.

Definitions and applications were identified and a general framework, addressing all the various stages, was drawn up. Its use was demonstrated in a small case study, the output of which is listed and explained. The process starts with retrieving the pertinent documents from the appropriate database. The last category is the visualization of the output. According to Sebastiani, text categorization belongs to supervised learning. However, our work depends on the unsupervised method rather than on the supervised.

For the evaluation of the clustering we used the SOM cluster validity technique. As in data mining, so in text clustering it is difficult to measure the performance of a particular clustering algorithm. In the case of text mining this generally depends on the data and the pre-processing techniques applied. The large number of features required when representing documents in a collection affects almost every aspect of a text mining system's approach, design, and performance.

In our experiments, we found that in general SOM is better at reducing dimensionality and at preserving the topology of input data. However, the full complexity of real world text data cannot be completely captured by known statistical models, and dimension reduction efficiency depends on the nature of the documents. In this thesis we present an algorithm called SOMMLTM that discovers appropriate clusters for Arabic-English texts. We describe examples and present experimental results that show that the new algorithm is successful. This technique is useful and applicable for many clustering algorithms other than SOM, but here we consider only the SOM algorithm for simplicity. This method for reducing dimensional is best applied with domain-specific knowledge and human intuition. The framework will be discussedin Chapter 5.

Chapter 3

Self-Organizing Map Technique and Neural Networks

Objectives

- To provide a background of Neural Network.
- To present NN Architecture.
- To discuss why the Self-Organizing Map (SOM) technique is important?.
- To present the self-organizing map algorithm.
- To apply an example on SOM.

3.1 Introduction

A neural network is an information-processing system based upon algorithms that seek to emulate the human brain's method of solving complex and large-scale problems at very high speeds. These algorithms organize complex patterns and store

and retrieve very large amount of information [78]. The following sections deal with the concept of neural networks, with an emphasis on SOM.

Alexander Bain (1818-1903), a famous British Utilitarian and early proponent of scientific psychology, first postulated the concept of a neural network in his description of memory as a set of nerve in 1873.

"If we suppose the sound of a bell striking the ear, and then ceasing, there is a certain continuing impression of a feebler kind, the idea or memory of the note of the bell; and it would take some very good reason to deter us from the obvious inference that the continuing impression is the persisting (although reduced) nerve currents from the past, the remembrance of the former sound of the bell" [79]. This asserted that abstract concepts such as memory could indeed be based upon physiological processes, and could therefore be reproduced if one had sufficiently high technology [80].

The artificial neural network consists of interconnected neurons, and these may be organized into groups. This was first proposed by McCulloch and Pitts [81] who gave their name to such groupings, the parameters for which were based upon their idea of a "threshold logic" unit. Such a unit also has input pathways which, to varying degrees, can active the neural groups [81]. Threshold logic was expanded by adding several input lines. As the inputs are not necessarily binary, they determined that the output terminal should be analog, contrary to common practice at the time. In order to fully capitalize on this flexibility, they also proposed the concept of weighted connections, making their neural network the first to be adaptable [82, 81].

Today's algorithms have a learning ability and this is originally due to Hebb, who in 1949 modified the way that input and output data behaved through the addition of "learning law" [78]. This is the combination of all the above factors that made it possible models researchers to develop the first artificial neural networks for use during the 1950s and 60s. The networks within these models were then called 'per-

ceptrons' [82].

Rumelhart, Hinton and Williams then developed, in 1986, a "backpropagation" method for training the models, and this was successfully employed in solving problems in a great many diverse fields. Its mechanism is based on a multi-layered feed-forward net trained by backpropagation [83].

A further step forward was made by Prof. Teuve Kohonen [84] who hypothesized that memory may have holographic characteristics in that similar cells grouped together in a field respond to particular stimuli, although each individual cell within a field will respond in a subtly different manner [85]. These groupings could be understood to be in layers and therefore could be described as maps, each one with its own basic characteristics but carrying within it a variety of cell types, grouped according to similarities [85]. Thus Kohonen invented a new approach, that of the SOM [86].

Ambiguity is inherent in language and many researchers have used a variety of unsupervised approaches to organizing words. However, they have all used only monolingual datasets [87], and this can be illustrated in the work conducted by Maa, Kanzakib, Zhangb, Muratab and Isahara [72] They mined newspapers and positional individual words on visible semantic SOM, where the distances between the words reflected their similarity, principally with respect to grammar. These monolingual maps were constructed for Japanese and for Chinese.

Yang and Lee [88] also used Chinese news documents but they constructed two distinctly different SOM, one which clustered words and another which clustered the news documents. In their work they categorized the word clusters and investigated the relationship between these categories and the centroids of clearly-defined clusters on the document SOM. If one accepts that each map represents a neuron, then they were investigating the connections between neurons or between layers in a network net.

Eyassu and Gamback [89] have applied SOM monolingual classifying Amharic onto a collection of news text for the task of retrieving the documents matching a specific query, then they have investigated document clustering around user queries using SOM.

Ping Li and Farkas [90] also worked bilingual language processing only on spoken Chinese and English data, were they present SOM model to learning the characteristics. In there work they can account for distinct patterns of the bilingual lexicon without the use of language nodes or language tags, it can develop meaningful lexical-semantic categories through self-organizing processes, and it can account for a variety of priming and interference effects based on associative pathways between phonology and semantics in the lexicon, and it can explain lexical representation in bilinguals with different levels of proficiency and working memory capacity.

Nikolaos and Helen [91] applied SOM and Latent Semantic Indexing for fully automated cross-language (English-Greek corpus) information retrieval which does not require any query translation. The clustering ability of SOM for the generation of multilingual semantic categories.

3.2 NN Architecture

A neural network (NN) is like a human brain, and is composed of a massive parallel collection of small and simple processing units where the interconnections form a large part of the network's intelligence. The human nervous system consists of a mass of neurons, interconnected by a network of synapses. The neurons can receive information from the outside world at several different points in the network. This information is called stimuli. The stimuli going through the network, then generating different responses in different neurons, which in turn send them, or new internal signals to neighboring neurons. These signals can be of varying strengths, depend-

ing on whether their situation is important or not. The signal causes a reaction in an individual neuron, either exciting or inhibiting it. If it is excited, that neuron passes the signal on to neighboring neurons, but if it is inhibited, it will not. This eventually produces a result, or reaction, from the network. For example, if a person drinks a cup of very hot tea, the nerves in his/her tongue register the hot tea, and then sends pulses to the nervous system. These stimuli pass through the network, resulting in the order for the nerves in the tongue to a void the hot drink. However, the network function is determined largely by the connections between elements [92]. See Figure 3.1 below.

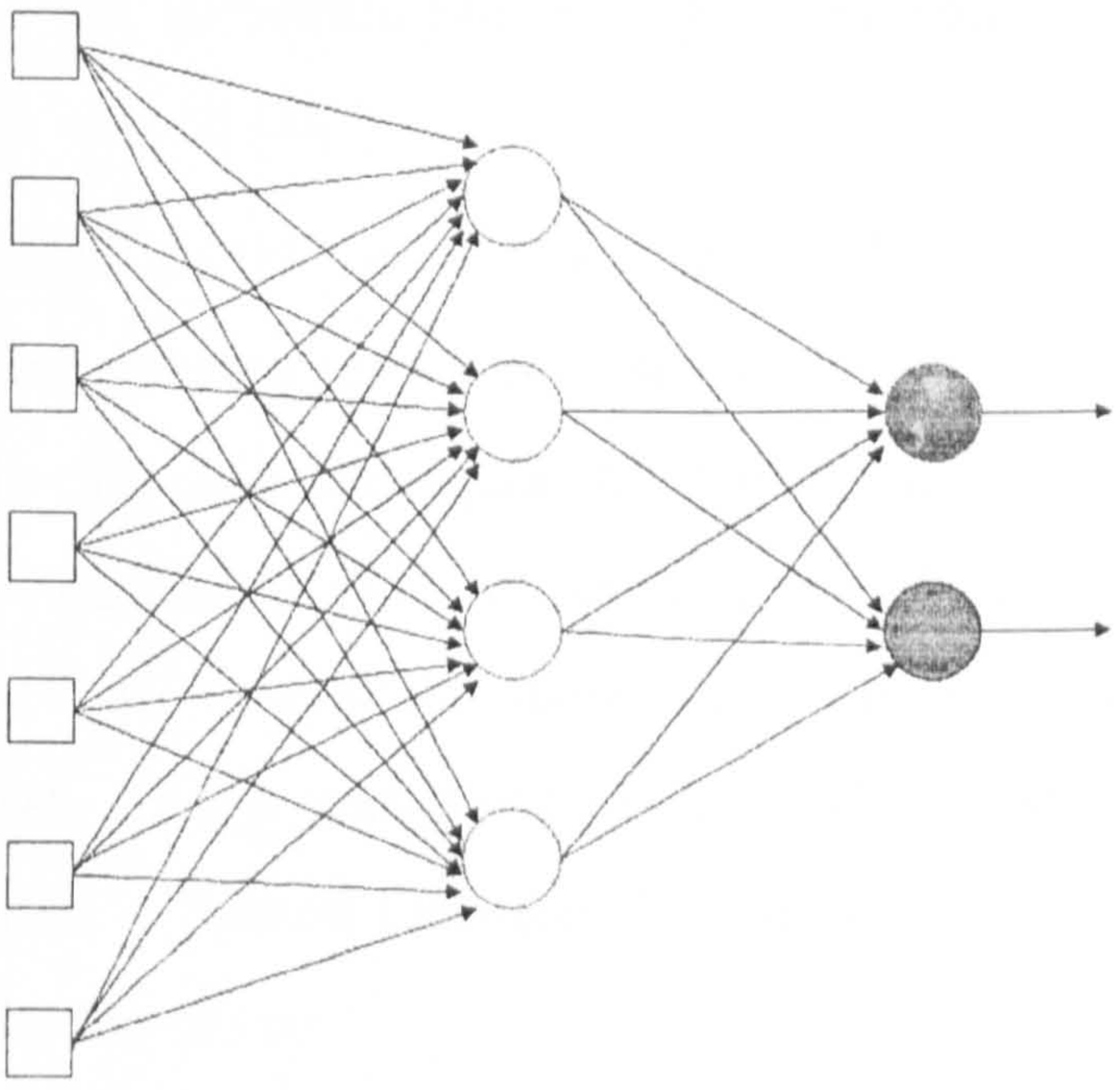


Figure 3.1: Fully Connected of Neural Network

Figure 3.1 above shows an example of neural network that consists of nodes (neurons) and weighted connections (synapses). These are arranged in a number of layers, usually an input layer, and output layer, and a number of hidden layers.

The input layer is the layer that is fed the data to be processed through the net-

work, while the output layer displays the result of the network. The hidden layers are where the actual processing of the data takes place. The nodes in the different layers are connected by a series of connections, each assigned a different weight.

A neural network (NN) is capable of learning and can make decisions. It is possible to differentiate between two major methods of NN: supervised and unsupervised. In the supervised learning method, a network learns with the help of training samples provided by a "teacher" or a supervisor who presents a training set to the network. In the unsupervised method, learning is needed, nor are training samples, the network learns by detecting similarities in the different input patterns (adapting without a teacher) i.e. a "reward function" [85]. Backpropagation learning is a supervised learning method [93].

Neural network models are the tools that support SOM. SOM is based on unsupervised artificial neural network models [17], and in the field of text mining. SOM employs automatic clustering techniques that have been specially adapted. They reduce high-dimensional input space vectors and reproduce the data onto a two-dimensional output space without eliminating any of the essential characteristics of the original data set. This project proposes that SOM can adapted for languages and faithfully utilized in Natural Language Processing (NLP), so that data can be mined from large bodies of text.

The SOM is neural networks, and are one of the most important techniques used in this thesis, employing unsupervised learning. The network is presented with input data, and is then allowed to organize itself, depending upon patterns that it recognizes within the input data. SOM consists of a two-layer neural network; an input layer, and an output layer. The results will provide for cluster analysis by producing a "topographic map of the input patterns in which the spatial locations (i.e. co-ordinates) of the neurons in the lattice are indicative of intrinsic statistical

features contained in the input patterns" [94]. After appropriate training iterations, the SOM clusters the similar input items by grouping them spatially close to one another through calculating the Euclidean distances between the input vector and the weight matrix. Such unsupervised, SOM is also called competitive learning [95]. The Kohonen SOM utilizes this approach, and it has been shown to be an effective algorithm for multilingual problems. For more detailed of algorithm used for multilingual text ming see Chapter 5.

3.2.1 Unsupervised Learning

SOM is clustering tools, i.e. they create a map on which similar groups of records are grouped close to each other, and natural boundaries between these groups are defined. Unsupervised learning means that this does not require a teacher who knows the correct classification or cluster (output) for an input pattern in the training set, and does not give class labels [96]. In the basic version, only one map node (winner) at a time is activated corresponding to each input. The goal typically is to generalize from specific examples presented from the past in order to learn for the future, giving more or less correct answers without intervention. Most simply stated, the goal is to find the natural structure inherent within the input data. There are a number of unsupervised learning schemes, including competitive learning and SOM, of which the most famous are the Kohonen networks, named after their inventor. SOM was further developed by Prof. Teuvo Kohonen in the 1980s and the first application of his SOM was speech recognition [97].

3.2.2 Supervised Learning

Feed-forward backpropagation neural networks are classification tools and an example of supervised learning. In this technique both input and the output (target) have to be specified in order to derive results of interest. As with SOM, the feed-forward

backpropagation technique can yield clusters on the basis of degree of similarity between documents. The latter technique was also employed in our study in order to compare with SOM network. Figure 3.2 displays the successful outcome of the feed-forward back-propagation technique. The convergence of the training curve to the goal line is interpreted as the similarity between the two input documents, while divergence in Figure 3.3 means the documents under trial are dissimilar. Although the feed-forward backpropagation network successfully classify the input documents, the short coming is that in a single run (training and testing) it only compares two documents at a time. Thus testing a set of large number of documents make this technique highly inefficient compared to SOM network. On this basis the SOM techniques is perfected in our study.

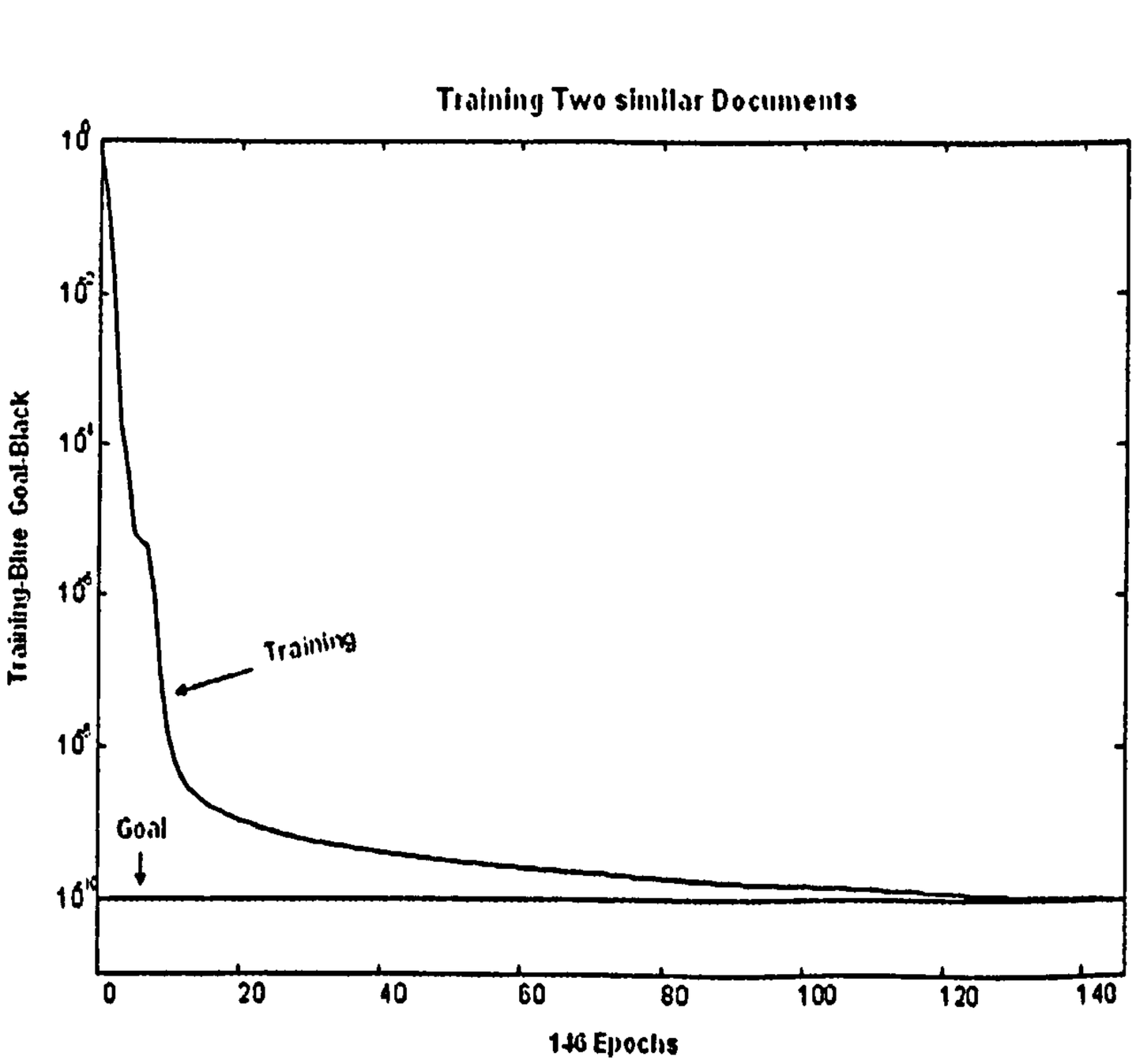


Figure 3.2: Illustrates the result achieved by feed-forward backpropagation network tested on a pair of bilingual similar documents. The maximum number of iterations (epochs) was set to 300 and with a learning rate of 0.1. The performance goal was met successfully and convergence in 146 iterations indicates a good learning machine.

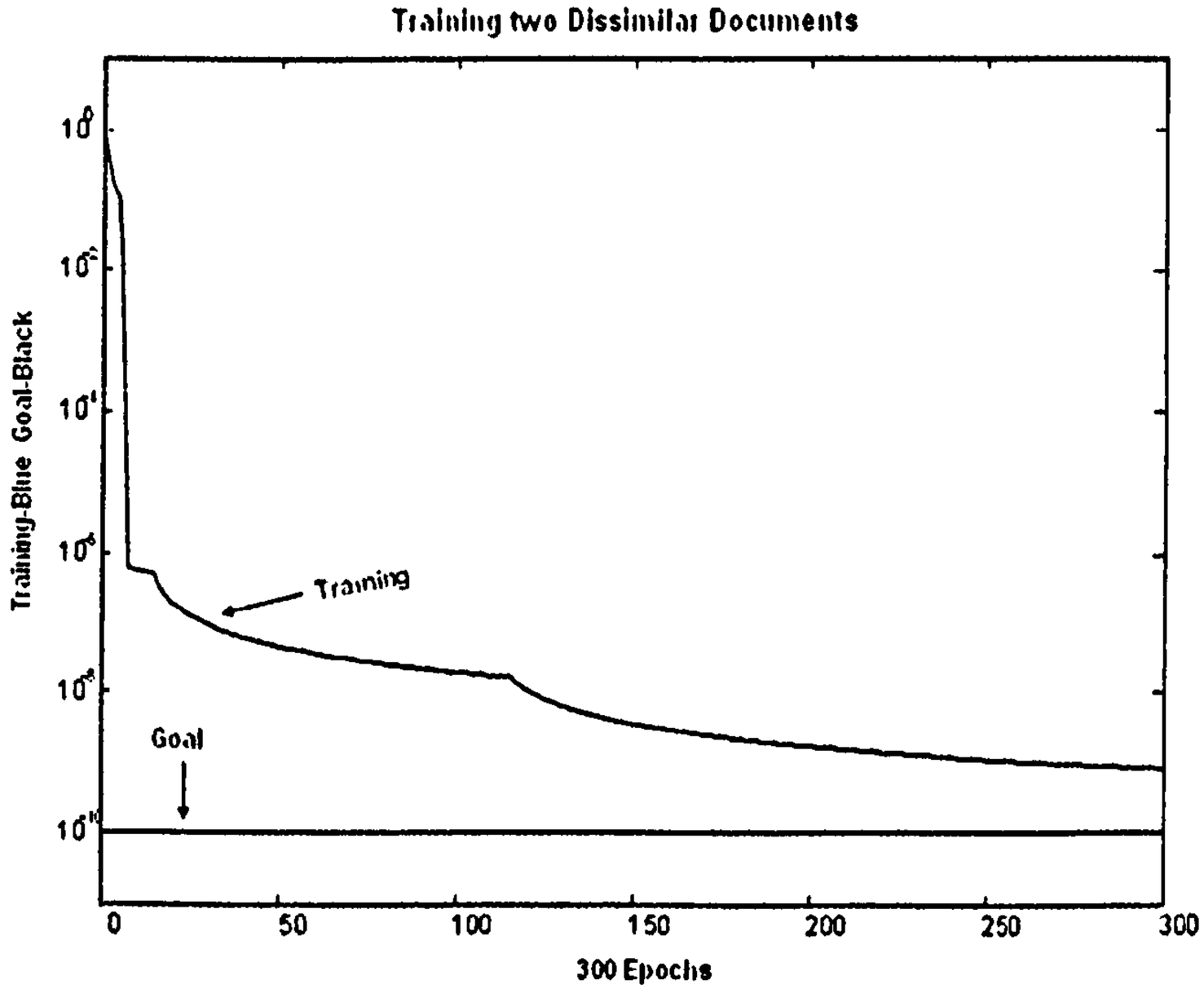


Figure 3.3: Same set of parameters as in Figure 3.2, but now employed on two dissimilar documents. Divergence is the indicator of dissimilarity between the two documents.

3.2.3 Kohonen Networks (SOM)

The objective of a network is that it provides a clear unambiguous result. However, it is possible that although all neurons in a net receive the same data input, the output classifications are duplicated [1]. This is despite the training that commands the input data to be classified into only one category, and so the architecture of the network needs to be modified in order that it provides only clear, unambiguous responses [68, 83].

The nodes also have lateral competition and the one with most activity is the "winner". This learning is based on the concept of winner neurons. An example of a network based on competitive learning is the SOM, as described below.

If there exists an ordering between the neurons, i.e. the neurons are located on a distinct grid, the SOM (the competitive learning algorithm) can be generalized. If not,

[1] Thus an item may appear in more than one category.

only the winning neuron is used but it may also be necessary to update the weight vectors in the neighbourhood of the winning neuron on the grid, and to decrease the size of the neighbourhood. Neighbouring neurons will step by step specialize to represent similar inputs, and the representations will become ordered on the grid. So competitive learning has neurons that have spatial structure but no neighbours and no geography, in contrast to Kohonen's SOM which has neurons with a spatial structure and neighbours that have a map distance between them; this is the core of the SOM algorithm [98].

3.2.4 Definition of SOM

SOM, also known as Kohonen maps are defined by [99] as follows: *"a data visualization technique which reduces the dimensions of data through the use of self-organizing neural networks"*, with the SOM algorithm arranging complex and very high-dimensional space onto a low-dimensional map grid.

Definition 3.1 as follows: *"SOM is a technique which is allowed to organize itself, and reduces a high-dimensional input space into two-dimensional maps, through learning can visualize data as a two-dimensional output grid".*

3.2.5 Components of SOM

Data

Organization of data is the heart of the SOM. This data may have numerous dimensions. Therefore, in order for it to be analysed, two processes must firstly be completed: reduction of dimensions and presentation of similarities. This latter should have a clear visual representation.

Weights

In relation to single dimension SOM places data in specific locations but it may also use weight vectors. Weight can also be thought of a neurons as they are integral to unsupervised networks, and so a "weight matrix" depends on the number of input vectors and output clusters. Weight matrix can be initialized randomly [2], or using the normalization technique.

3.2.6 Self-Organizing Map (SOM)

The basic SOM can be visualized as a sheet-like neural-network array (see Figure 3.4) consisting of an input layer and an output layer in which every input node is fully connected to each output node. The SOM is based on unsupervised, competitive learning [50], and provides a topology preserving mapping from high-dimensional space to map units. Neurons (nodes) usually form a two-dimensional grid and thus the mapping is reducing from high-dimensional space onto low-dimensional space. The property of topology preservation means that the mapping preserves the relative distance between the points through learning processes. The nodes that are near each other in the input space are mapped in the same neuron in the SOM, which thus serves as a cluster analyzing tool of high-dimensional data. Also, the SOM has the capability to generalize, meaning that no teacher is needed to define the correct output, or for the node to recognize or characterize inputs it has never encountered before. A new input is assimilated with the map unit it is mapped to. Thus the network is presented with input data, and is then allowed to organize itself, depending upon patterns that it recognizes within the input data.

The learning process, also called competitive learning, is distinguished by a competition among the neurons. Unlike supervised learning, in which several nodes can fire at the same time, in unsupervised learning all the output neurons compete to be a single neuron, i.e. "the winner". After appropriate training iterations the in-

put patterns form groups of similar data. SOM techniques are a major example of applications that use unsupervised learning [94].

The mapping tries to preserve topological relations so that it can work on monolingual and multilingual languages. This clustering of data, provides a two-dimensional display of the input space that is easy to visualize, making this technique more effective than others. A successful application of SOM is the example of [17] where this algorithm for has been applied to the Chinese and English languages. SOM performs well in clustering data, which is then easily evaluated for quality and for the strength of similarity between the objects [99].

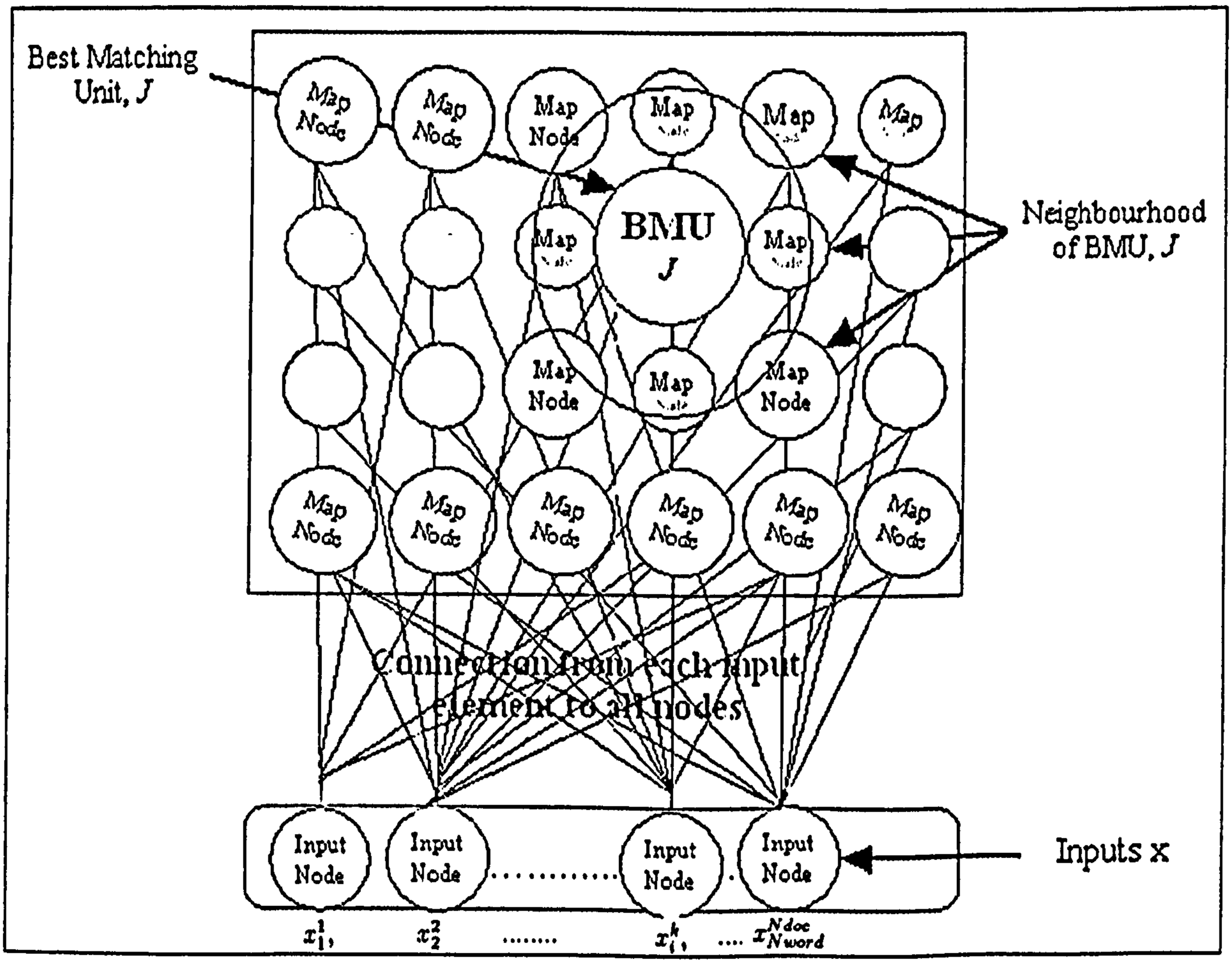


Figure 3.4: which is adapted from Honkela [1], shows the basic architecture of SOM. The input $X = (x^1_{1-Nword}, x^2_{1-Nword} \cdots, x^k_{i-Nword}, .., x^{Ndoc}_{1-Nword})$ is fully connected to all nodes in two-dimensional 4X6 grid. Each node is visualized as a circle on the grid, the BMU (winner) is J. The neighbourhood of the BMU is N_J. Where Ndoc and Nword are number of documents used and number of words in the largest document

One of the significant outcomes of this study is that SOM can be a dynamic option to customary clustering techniques. Owing to the above, there are now four important reasons why SOM technique has been selected. Firstly, The SOM is popular among researches in a wide variety of fields. Secondly, SOM does not rely on any statements based upon statistical tests. Thirdly, the SOM is able to deal with data that do not have regular multivariate distributions whereas other statistical clustering methods need such distributions. Fourthly, the quality test decides whether more training is needed or if the clusters are acceptable. In problems related to text mining the scarcity of suitable tools for the analysis of large amounts of information is present. One possible tool for this purpose is the SOM, which has been applied to a wide range of problems, and in some cases, specifically to multilingual language problems [17]. Finally, the visualization of the results is a very strong attribute of the SOM, therefore, based on these advantages, SOM will be used for this work. On other words, using the SOM technique in order to test a set of large number of documents make this technique highly accurate, and proves the efficiency of the network.

3.2.7 Self Organizing Map Algorithm

SOM works on topological space, in which each input node (vector) $x^k_{i\_Nword}$ connected to the each output node (cluster). Input nodes (neurons) are connected to the output through the weight matrix w_ij. The weight matrix has $Nword$ x m dimensions where m is the number of desired clusters. Usually each element of the weight matrix is initialized randomly. The grid formed by the input and the output nodes can be either rectangular or hexagonal in structure as can be seen in Figure 3.5. The column vector J ($1 <= J <= m$, BMU), from the rest of the constituent column on the basis that it mostly resembles with the input pattern (typically the square of the minimum of Euclidean distance). Not only the BMU

is updated but its meighbourhood as well. The neighbourhood is defined by the integer R=0,1,2,3,.... For R=0 only BMU (the winner) is updated. For linear grid with R=1 the neighbour node on the right and the one on the left are also updated, similarly for R=2,3.. can be extended. The processes of updating (learning) the weight matrix is controlled and accelerated by a parameters called as the (learning rate, α). The parameter α itself reduces with each iteration (epoch) according to the set conditions, e.g. $\alpha(t+1)$=$\alpha_0.\alpha(t)$ where $0 < \alpha_0$, $\alpha(t) < 1$.

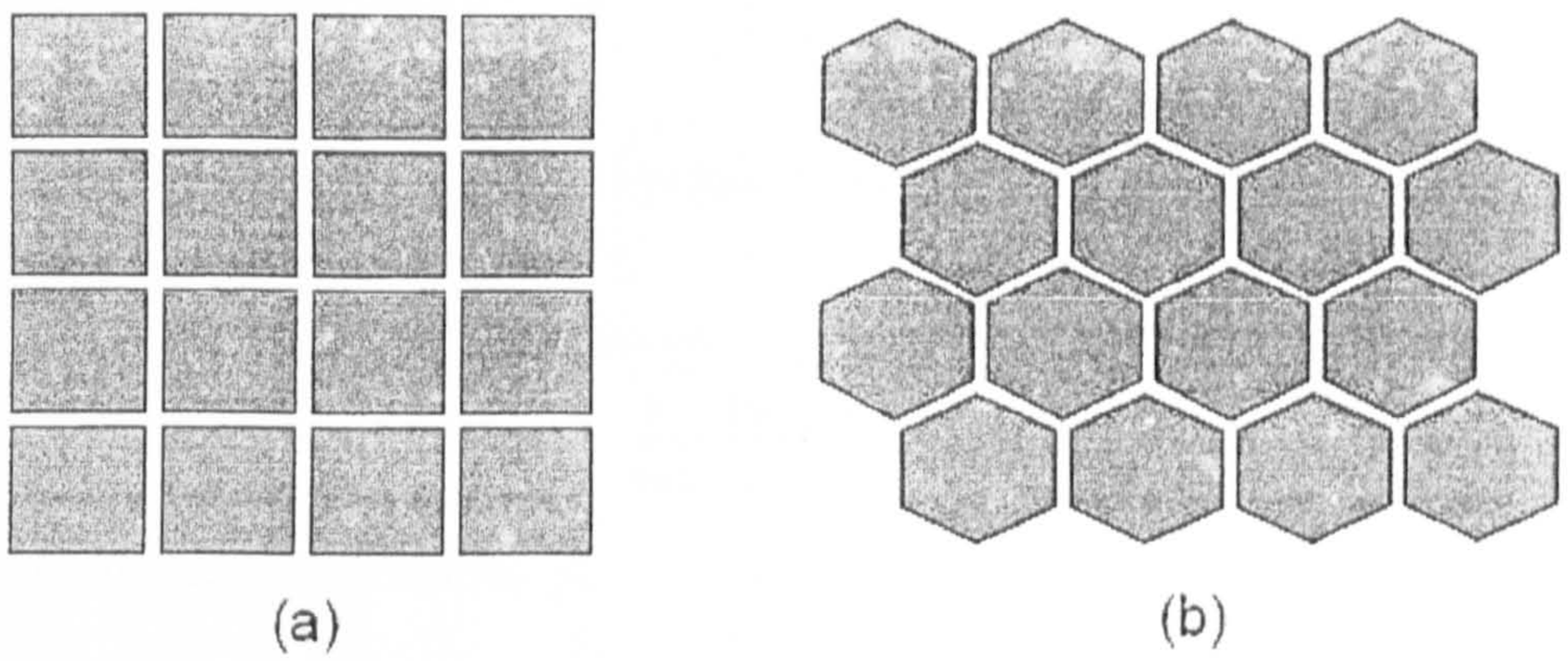


Figure 3.5: which is adapted from Kohonen [2].(a) Rectangular grid (size 4x4)and (b) Hexagonal grid (size 4x4)

The SOM algorithm performs a series of steps and these are repeated until no further changes in the output data can be discerned or until some finishing criterion is met. However, the algorithm must first be trained and this needs to be included in the steps in order to determine the vectors that will identify the clusters [83, 100, 101]. Below are the necessary steps involved in each SOM algorithm.

Step 1 Set parameters of topological neighbourhood radius ($R(t)$) and learning rate $\alpha(t)$, where R can be a variable parameter [99].

Step 2 Selection of a training input data vector; The data set consists of data

vectors such as X.

$$X = (x^1_{1-Nword}, x^2_{1-Nword} \cdots, x^k_{i-Nword}, .., x^{Ndoc}_{1-Nword}) \tag{3.1}$$

where i=1,2,...., Nword

Each node (neuron) in the map is associated with input vector such as w_{ij}:

$$W = [w_{i1}, w_{i2}, ..., w_{ij}] \tag{3.2}$$

Step 3 Find the neuron j with the smallest Euclidean distance to the training data set x^k_i, calculate as:

$$D^k_{(j)} = \sum_{i=1}^{Nword} (w_{ij} - x^k_{i-Nword})^2 \tag{3.3}$$

j=1,....,cluster, for each neuron j.

Where D^k_j is the minimum value of Euclidean distance of all neurons.

Step 4 Selection of points for the BMU 'J' i.e. $(minD^k_j)$ for each k.

Step 5 The weight matrix is updated. The BMU and its topological neighbours are moved closer to input vectors in the output space. The update is according the formula:

$$w_{il}(\text{t+1}) = w_{il}(\text{t}) + \alpha(t)[x^k_{i-Nword} - w_{il}(\text{t})] \tag{3.4}$$

for $l = f(R)$.

Step 6 Update the learning rate $\alpha(t)$. Some of the common form of $\alpha(t)$ are given below:[2mm]

1. $\alpha(t+1)=ae^{-bt}$, where $b > 0$, $0 < a < 1$ or

2. $\alpha(t+1)=at^{-b}$, where $b <= 1$, $0 < a < 1$ or

3. $\alpha(t+1)=a(1-bt)$, where $0 < b < (maxt)^{-1}$, or

4. $\alpha(t+1)=a\alpha(t)$, where $0 < a < 1$

Also, reduce the neighbourhood function either considering $R(\alpha)$ or $R(t)$. R can be considered as a neighbourhood "function" i.e. $R(\alpha)$ or $R(t)$. For example:

In the beginning a large neighbourhood can be selected i.e. say R=3 but with the passage of discrete time t, $R(t)$ reduces: 3,2,1,0. Similarly, $R(\alpha)$ changes with reduction in $\alpha(t)$. The quality of thc given map may also be determined through measuring be calculating the average quantization error (qe), given by [1]:

$$qe = \sum_{k=1}^{NdocNword} \sum_{i=1} \parallel x^k_{i-Nword} - J \parallel \tag{3.5}$$

It simply gives the average distance of the difference of the BMU from set of input vectors $x^k_{i-Nword}$.

Step 7 The training process ends by reaching a certain limit in weights or where the changes are stable, otherwise continue with a return to step 2.

3.3 Applied Example

Kohonen applied his SOM to a variety of interesting problems. This technique is also applied to solve the classic problem of the travelling salesman. Below is a simple application of SOM.

Example: For illustration, we have applied the algorithm of Kohonen SOM on the set of following four input vectors to obtain only two clusters [83].

$X = [x_1$=(1,1,0,0); x_2= (0,0,0,1); x_3= (1,0,0,0); x_4= (0,0,1,1)].

Set the maximum number of clusters to be formed is m = 2.

Suppose the learning rate is:

$$\alpha(0) = 0.6$$

$$\alpha(t+1) = 0.5.\alpha(t)$$

With only two clusters available, the neighbourhood of node J is set so that only one cluster updates its weight at each step (*i.e.*, $R = 0$).

Initial weight matrix:

$$\begin{bmatrix} .2 & .8 \\ .6 & .4 \\ .5 & .7 \\ .9 & .3 \end{bmatrix}$$

Step 1 *Begin training*

Find the neuron with the smallest Euclidean distance to the training data set X, defined as:

$$D_{(j)}^{k} = \sum_{i=1}^{Nword} (w_{ij} - x_{i-Nword}^{k})^2$$

D_J is the winner neuron.

Step 2 *For each input vector* $x_{i-Nword}^{k}$, *perform Steps* $2-4$

Step 3 Change the winner weights and the weights of neighbouring neurons until they are close to the input vector inthe output space. This learning process can be defined as:

$$w_{il}(\mathrm{t+1}) = w_{il}(\mathrm{t}) + \alpha(t)[x_{i-Nword}^{k} - w_{il}(\mathrm{t})]$$

Then it will repeat these steps by using the algorithm until all vectors are completed.

Step 4 *Finally the learning rate is reduced.*

$$\alpha = .5(0.6) = .3.$$

The adjustment procedure for the learning rate is modified so that it decreases geometrically from 0.6 to 0.01 over 100 iterations. In this case, after completing these iterations, then the weight matrices appear to be converging to the matrix as below:

$$\begin{bmatrix} 0.0 & 1.0 \\ 0.0 & 0.5 \\ 0.5 & 0.0 \\ 1.0 & 0.0 \end{bmatrix}$$

Mathematically, the first column is the average of the two vectors placed in Cluster[1] and the second column is the average of the two vectors placed in Cluster[2], i.e. vectors 2 and 4 are similar and placed in the same Cluster[1]. On the other hand, vectors 1 and 3 are similar and placed in the Cluster[2].

We can relate these vectors to physical entities for clarity.

x_1 =(Apple, Banana, 0, 0).

x_2 =(0, 0, 0, Orange).

x_3 =(Apple, 0, 0, 0).

x_4 =(0 , 0, Grape, Orange).

Actually, after the network has been trained with the input descriptions, a grouping is made according to the similarity of vectors. Vectors (x_1) and (x_3) are similar to each other because they have "Apple" in common, so they are placed into Cluster[2]. In contrast, the vectors (x_2) and (x_4) are similar to each other because they have

"Orange" in common, so in this case they are placed into Cluster[1].

3.4 Data Visualization Using SOM

The basic visualization of the SOM in freely specified coordinates, for example the input space (of course, in high-dimensional space). This function has many options and is extremely flexible. Needless to say, functions and their usage have already been introduced. Data visualization techniques using the SOM can be divided to three categories based on their goal:

1. visualization of clusters and shape of the data: U-matrices and other distance matrices.

2. visualization of components / variables: component planes, scatter plots.

3. visualization of data projections: hit histograms, response surfaces.

Prior to creating a SOM, it must be visualized in order to interpret it. Moreover, unified distance matrix, or U-matrix [102], is the most common way for visualizing SOM. U-matrix map is created by calculating the average of the distances of a reference vector in a node to that of its neighboring neurons, and store them in a matrix, that is, the output map, which then can be interpreted. This average is placed at the appropriate coordinate on the matrix. The shape of the matrix is dependent upon the neighbourhood topology, i.e. rectangular or hexagonal. The distance values are also displayed in color when the U-matrix is visualized. Hence, with peaks and trough. Peaks, represent great distances between reference vectors, are displayed on the map, while trough areas indicate similarities among the neurons. Figure 3.6 displays U-matrix maps in grayscale.

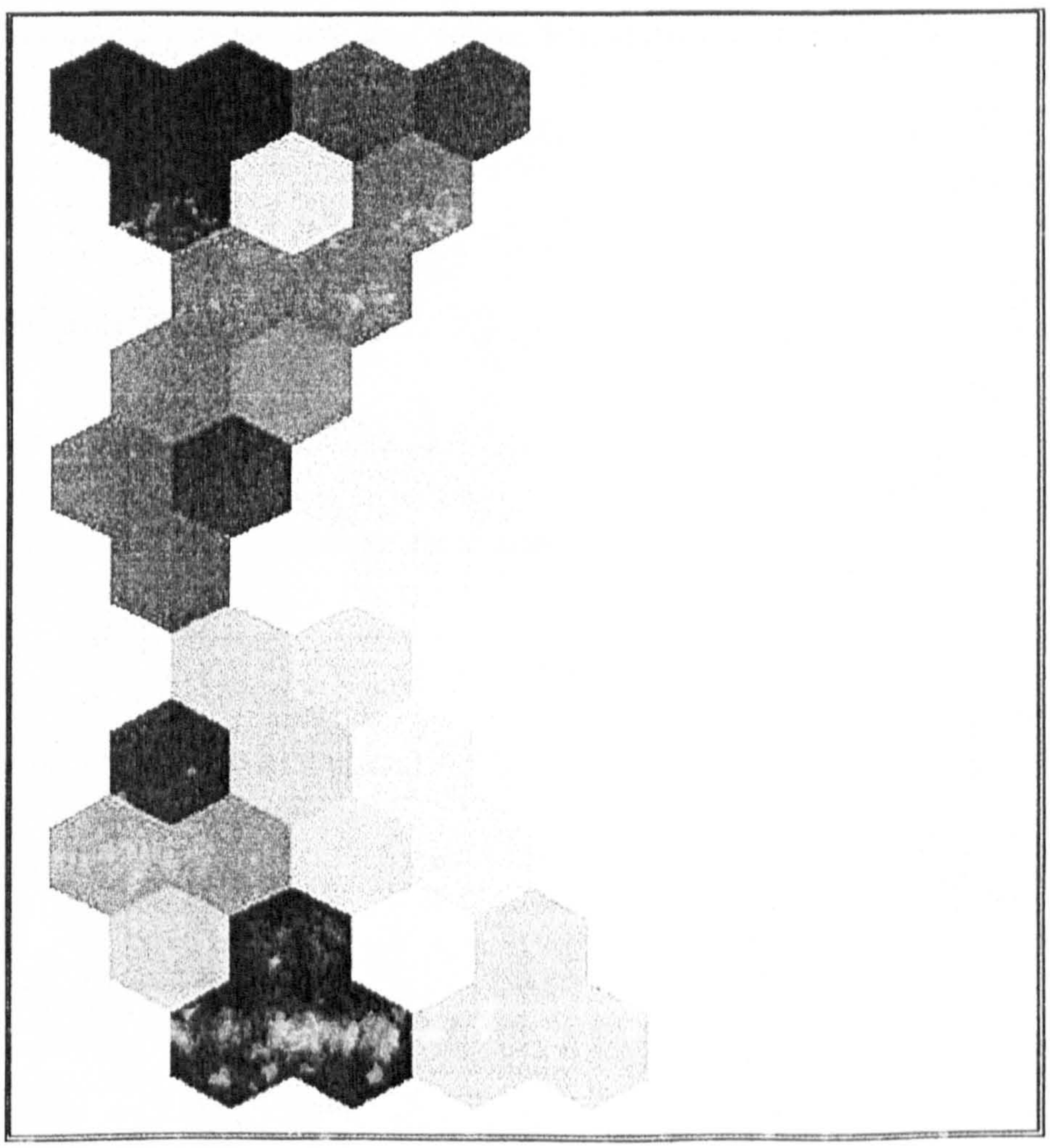


Figure 3.6: U-matrix Map in Grayscale

3.5 Application of Self-Organizing Maps

Many studies have been applied on SOM. According to [103, 104, 105], fields ranging from engineering sciences to medicine, biology, NLP, and economics have been studied through the application of SOM, spanning a great many years.

Researches have collected a comprehensive list of 5384 scientific papers that are related to Kohonen's algorithms, and below is a short summary of the most common areas in which SOM have been applied:

1. Speech recognition.

2. Image processing.

3. Engineering applications.

4. Diagnostics in industry and medicine.

5. Mathematical problems.

6. Data processing (NLP).

7. Financial analysis.

3.6 Summary

The concepts of neural networks as maps that are ordered in layers was first proposed by Kohonen [2]. His work built upon developments by McCulloch and Pitts who envisaged an artificial neural network of interconnected neurons. They saw that inputs could either excite or prevent a neuron or node from becoming active, and that the output need not necessarily be binary, it could be analog.

The above processes are performed by algorithms but these need training to make them specifically appropriate to a task, and in this regard, the work of Hebb is significant.

Network function is a product of neuron connections and the strengths of the signals that pass between them. These are organized so that the neurons lie in layers and the signals are weighted. These concepts support SOM, which are usually applied so that a researcher has only one input layer and one output layer. The clustering that takes place within the SOM reduces the dimensionality of the data. The position and number of output neurons on SOM is determined through competitive learning, where the "winners" are the neurons that most closely reflect the "topographical" features or patterns within the data. Depending on our objectives, the input vectors may be weighted and this manipulates the neighbourhood of a neuron so that learning rate and the neighbourhood radius are reduced, making the output more clearly defined. The visualization of the output is most commonly performed with a unified distance matrix.

Learning can be either unsupervised or supervised and it is the former that is most often employed in SOMs; it does not need a teacher or class labels, and uses the competitive approach mentioned above. Supervised learning, on the other hand, utilizes feedforward and backpropagation methods where the input and output fields

[2] Where data is ordered or clustered in zones of similarity. His works enabled researchers to reveal the patterns inherent within large dataset as a visual representation on a SOM

are specified in advance. Unfortunately, this is only most practicable on small data sets.

We must set the parameters for the learning rate and the radius of the neighbourhoods , and select an algorithm for training. The training process has several steps, these have been identified in this chapter. When, regardless of further iterations of the algorithm, no further changes in the output feature map are discernable, we may conclude that the patterns being sought are now fully revealed. Updating can be determined through Equation 3.4.

The goal of this work is not to study or introduce the mathematical and statistical properties of the SOM. But, to consider SOM as a model for natural language interpretation, and its application using in natural language processing (NLP), especially in information retrieval and text data mining of large collections of documents. It does not require huge amount of memory, just the prototype vectors and the current training vector [106], as depict in chapter 5.

In general, most research projects utilizing SOM has been concerned with numerical data, and indeed this research is also an investigation into the ability of SOM to efficiency identify statistical features, but within natural text. This is significant as the application of SOM in NLP task is principally the identification of numerical features. However, this chapter has also demonstrated how SOM is capable of clustering textual data that has been converted to numeric data making it of particular use to researchers in the mining of text data. Most researchers have used SOM on multilingual rather than monolingual datasets and there is no SOM research specifically available on the Arabic language. Indeed, to our knowledge, there are very few text-mining studies on Arabic specialised texts as there has been little academic research. Therefore our work should offer researchers an ever-widening range of applications.

Chapter 4

Arabic Language Structure

Objectives

- To provide a background to the Arabic language.
- To discuss the complexity of Arabic morphology.
- To present the stemming technique for the Arabic language.

4.1 Introduction

It is well known that families have different groups and each group has it's own branches, and to a greater or lesser extent, this is also true for languages. For example: "group of languages spoken in northern Africa and the Middle East that constitutes one of the branches of the Afro-Asiatic (formerly Hamito-Semitic) language family. (The other branches are Egyptian, Berber, Cushitic, and Chadic). The Semitic languages are divided into four groups: (1) Northern Peripheral, or Northeastern, with only one language, ancient Akkadian; (2) Northern Central, or Northwestern, including the ancient Canaanite, Amorite, Ugaritic, Phoenician and

Punic, and Aramaic languages as well as ancient and modern Syriac and Hebrew; (3) Southern Central, including Arabic and Maltese; and (4) Southern Peripheral, including South Arabic and the languages of northern Ethiopia" [107]. We have selected these languages in order to provide a context for the focus of this study, the Arabic language. The Arabic language is the mother tongue for over 250 millions people in the 21 Arab countries of North Africa and the Middle East. It is one of the Semitic languages and, although with significant differences, has many characteristics in common with other natural languages such as Indo-European languages. Furthermore, Arabic is used as a second language in Islamic countries [1]. Although there is a single form called classical Arabic, there are many regional variations and dialects.

This chapter examines the structure and lexis of the Arabic language, Natural Language Processing for Arabic, morphological analysers in general, and types of Arabic Morphological Analysers.

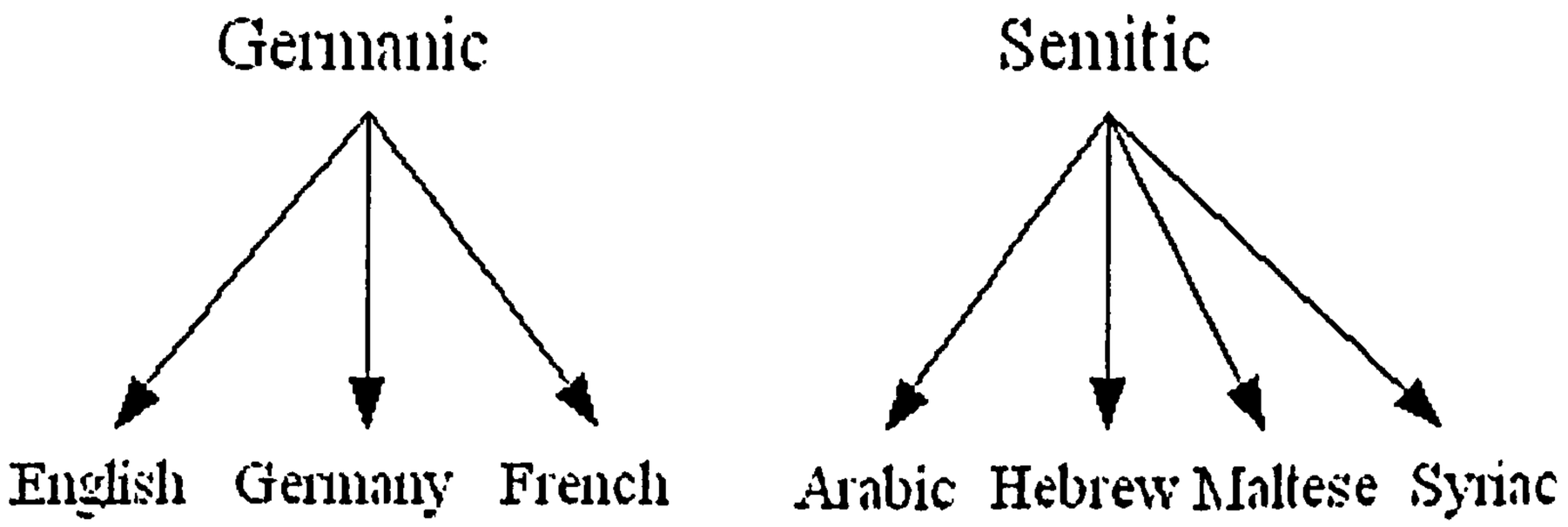


Figure 4.1: Language Families of Which English and Arabic are Member

Figure 4.1 shows that the English, German and French etc, are derived from Germanic groups. In contrast to English which is heavily influenced by "Latin". Arabic has itself been an influence many other languages, especially in the Islamic world

[1]Therefore about 300 millions people understand Arabic [108, 25] and so it is recognized as one of the six official languages of the United Nations. It is also of geo-political importance as approximately 65% of the world's oil reserves [109] are located within Arabic-speaking countries

and even some European languages.

4.2 Comparison of Arabic and English Processing

Arabic differs from English in various ways that affect text mining systems. In the context of this research, Arabic has many aspects in common with English, including syntax and the use of synonyms and homographs. Thus Arabic can be classified into familiar categories, principally noun (اسم, *sm*), verb (فعل, *fEl*) and particle (حرف, *Hrf*). However, further sub-divisions are possible, and in Arabic syntax, a single word can constitute a sentence; for example, (*Akla*, اكل)(he ate) or even the same word with the pronoun (ت) as (*Aklto*, اكلت)(she ate), and therefore this research must first identify the Arabic under consideration. Arabic has three different forms: (1) Classical Arabic, which is the language of the Holy Qura'n; (2) Modern Standard Arabic (MSA), which is the language of the newspapers; and (3) Colloquial Arabic (Al-Amỹh, العَامية), which is the language used everyday in oral communication. MSA is derived from classical Arabic and, being the language of the holy Qura'n, is understood across the Islamic world. MSA is widely employed in the media and so many people will listen to and read MSA but conduct conversations in local dialects.

There has been relatively little computational linguistics research into Arabic and there are not many fully electronic databases. There are two probable reasons for this: the complexity of the language, the lack of computer tools and models. Therefore there has been too little computational data to improve upon the models already available for Arabic.

Some researchers have used transliteration; converting Arabic characters to a Roman script allows researchers to use other computational models. In this regrad, the Unicode standard was employed by Gate [110, 111] and Nooj [112]. Interest in

mining Arabic text increased particularly following the September 11/2001 attacks in USA. The reasons were twofold; to improve security and to build bridges through a better understanding of the Arabic world.

4.3 Arabic Writing System

Arabic letters, unlike Roman ones, can change their shape according to their position in a word: initial, medial or final. Additionally, they may be joined together or kept separate, depending on th preceding and succeeding characters. Furthermore, Arabic is written from right to left and carries no capital forms.

The Arabic alphabet consists of 28 letters which are called (Hrwf AlHjA', حروف الهجاء). Of the 28 letters, 25 represent consonants and there are three letters which appear in different shapes called (Hrwf AlElp, حروف العلة) or long vowels (/i:/,/a:/,/u:/) [30], as shown in Table 4.1 below:

Table 4.1: Different Shape of Characters

Hamza	أ	أ	إ	أي	أو	أـ
Example	بهاء	ألم	إيمان	بريء	شؤون	عائلة
Transliteration	bhAa	> *lm*	< *mAn*	bare'a	shuwn	EAlah
Meaning	beauty	pain	faith	innocent	affairs	family

Alif Magsurah	ى	ي
Example	شكـوَى	شخصـي
Transliteration	shakwY	shaxsy
Meaning	complaint	personal

Arabic words are often organized in a Verb-Subject-Object (VSO) order while English is organized in a Subject-Verb-Object (SVO) order. Most Arabic words mor-

phologically resulting from a list of roots, which are usually made up of three consonants.

4.3.1 Complex Morphology

In comparison to the English language, Arabic has rules for words from which various other words can be derived eg.(hitting, *Drba*, ضرب)as shown in Table 4.2 .

Table 4.2: Complex Morphology

ضَارب	ضَاربَا	ضربَا	يضرب	ضرب
DAriba	DArbAF	DrbA	yDrbu	DrbA
مضَاربين	مضَاربون	يضَارب	مضرب	مضَارب
muDAribyn	muDArbwn	yuDArib	mDrab	muDArib

Fully-formed words are derived from root words, as already mentioned only a few characters long. However, some derivations eliminate one or even two letters of the root word. This can make identification of the original root fairly ambiguous, as shown in Table 4.3 below.

Table 4.3: Derivations of Word

Arabic	Translation	Arabic	Meaning	Transliteration
يدي	my hand	يد	hand	yado
أخو	my brother	أخ	brother	Axo
دمي	my blood	دم	blood	damo

Arabic words consist of roots that obey certain patterns, and these patterns allow for a variety of stems. Arabic syntax permits the use of multiple prefixes and suffixes, including articles, conjunctions and particles; up to four prefixes may be added, and as many as six suffixes, although they are mostly single characters. In contrast, roots are usually two to four characters, occasionally five [113]. Table 4.4 below

shows how these stem-to-root patterns, or templates, work for three-letter words [114].

Table 4.4: Some templates for three letters from root(ktba,كتب)

الأوزان	فعل	فَاعل	فعّال	مفعول	مفعله	مفَاعل	مفَاعيل
Example	كتب	كَاتب	كتَّاب	مكتوب	مكتبه	مكَاتب	مكَاتيب
Transliteration	ktba	kAtib	kitAb	maktwb	maktbah	mkAtib	mkAtib
Meaning	write	writer	book	letter	library	offices	letters

This ability to use prefixes and suffixes to modify the meaning of a root is common to both Arabic and English, i.e. they are both concatenate languages. However Arabic root derivation is more complex and extensive, and it allows for greater flexibility in meaning. It is also allows for changes in the sequence of letters that make up the final word. This is not possible in English, which is a mainly concatenate language, as are German and Spanish. Thus the templates are central to Arabic as they control suffixes, prefixes, which together can change the appearance of a root to a significant degree.

Arabic and English are both inflective, i.e. words must agree in case, number, gender, person, tense etc. Nouns, verbs and adjectives are all subject to inflection although this does not alter the grammatical category of a word (singular, mufrad, مفرد), (dual, muvnA, مثنّى) and (plural. jamE, جمع). Arabic is a rich language, due to it's cultural and religious background, but due to its emphasis on suffixes and prefixes, Arabic is a rule-based language. Arabic is more inflective than English and is a regular language in comparison to English. Thus, Arabic templates are richer and more complicated but regarding text mining it is governed by rules, that can be read by an algorithm. Thus Arabic is a challenging language for a number of

reasons [115, 116, 117, 118].

4.3.2 Orthography with Diacritics

Arabic orthography is highly variable. Orthographic signs, or diacritics, are the non-alphabetic characters added above or below the consonant letters to make the reading of the words less ambiguous and more phonetic; the form of certain letters in Arabic script allows for certain combinations of characters to be written in different ways and therefore orthographic signs add clarity. For example, the Holly Qura'n is always fully "signed" to avoid any misreading [119].

4.3.3 Broken Plurals

Broken plurals, analogous to irregular nouns in English plurals (e.g. "woman-women"), are very common in Arabic, except that they often do not resemble the singular form as closely as irregular plurals resemble the singular in English. Because broken plurals do not obey normal morphological rules, they are not handled by existing stemmers such as the Buckwalter stemmer. Table 4.5 shows irregular plurals:

Table 4.5: Broken Plurals

	Arabic	Transliteration	Translation
Singular	مدرسة	mAdrsap	school
Plural	مدَارس	mAdAris	schools
Singular	طفل	tifil	child
Plural	اطفَال	AtfAl	children

4.3.4 Short Vowels

Arabic language has distinctive vocal sound system, which is highly different than English language."It includes a number of distinctive guttural sounds (pharyngeal and uvular fricatives) and a series of velarized consonants (pronounced with an accompanying constriction of the pharynx and raising of the back of the tongue)" [107]. The Arabic language has short and long vowels that change the pronunciation and sometimes change the meaning. There are three short and three long vowels (/a/, /i/, /u/ and /a/, /i/, /u/). Arabic words always start with a single consonant followed by a vowel, and long vowels are rarely followed by more than a single consonant. Clusters containing more than two consonants do not occur in the language [107]; grammatically they are required but are omitted in written Arabic as shown in Table 4.6.

Table 4.6: Short Vowels

Word	Meaning
قلم ⇐ قلم	Pen
qalma ⇒ galm	Transliteration

4.3.5 Synonyms in Arabic

As with English, good quality written styles of Arabic contain the use of synonyms, and there is a great variety of them.

As a concatenate language. Conjunctions, articles, particles and pronouns, amongst others, are added as prefixes and suffixes in order to satisfy grammatical rules. Examples are shown in Table 4.8 below, adapted from [120].

Table 4.7: Synonyms

Arabic	وهب	أعطى	منح	بذل
Transliteration	whaba	AEtA	maHa	bzla
Meaning	grant	grant	grant	grant

Table 4.8: Some examples of clitics

Conjunctions	**Pronouns**
and (*w*, و)	his (*ha*, ـه)
like (*k*, ك)	your;singular (*ka*, ك)
then (*f*, ف)	your;plural (*kum*, كم)
Articles	their (*hum*, هم)
the (*Al*, ال)	her (*ha*, هَا)
and the (*w* – *al*, وَال)	my (*y*, ي)
Particles	your;dual-male(*huma*, هما)
to (*l*, ل)	your;dual-female(*huna*, هن)

4.3.6 Clitics

Arabic has a set of attachable clitics to be distinguished from inflectional features such as gender, number, person, voice, aspect, etc. These clitics are written attached to the word and thus increase the ambiguity of alternative readings [121]. Sproat [122] defines a clitic as "a syntactically separate word that functions phonologically as an affix". These are distinct from inflective prefixes and suffixes but cannot be fully pronounced as separate words and so become phonologically attached to be beginnings or endings of words. Arabic has attachable clitics. Several English clitics are reduced variants of non-clitic words [123] like am ('m)and not ('t), but many clitics in Arabic are usually equal to stop words and closed class words in English. For example, the grammatical conjunction (*wa*,و) is a commonly used letter, which is equivalent to the English conjunction 'and', (*f*,ف) equivalent to 'then' and the definite article 'the' equivalent to (*Al*,ال) in Arabic are always attached to the beginning of the following word, for instant the phrase 'and the book' is equivalent to one

word in Arabic (*wa – alktab*, وَالكِتَاب). [124] Observes that the commonest closed-class words in English are used predominantly in relation for reference (e.g. that, it, a) and to syntactic extension using propositions and conjunctions (e.g. from, for, of, on, to,..). The same is true for Arabic, but it becomes more apparent when splitting off the clitics.

The clitics include some of prepositions, conjunctions, determiners, possessive pronouns and pronouns. But there are some proclitic attached to the beginning of a stem and some enclitics attaching to the end of a stem.

In contrast to English, adjectives in Arabic mirror the definiteness of the noun, so the phrase 'the Arabic story' has the literal translation of ('the-story the-Arabic', *Al–qiSah Al–Erabyah*, القصة العربية) in Arabic. Short vowels that are represented as diacritics marks in Arabic are also omitted from the majority of texts except for example children's and grammar books, causing some different words to appear exactly the same making word sense disambiguation more difficult. Although the diacritic marks reduce the ambiguity, an average reader should be able to determine the correct word from context. Diacritic marks are usually used when there is a high fear of ambiguity in determining a word sense or grammatical case and for marking indefiniteness, which corresponds to the use of 'a' and 'an' in English. Arabic does not mark indefiniteness with a separate word, nunation (*tanwin*,تنوين), a suffix sound /n/ indicated with a inflectional (short) vowel on a word twice, is used at the end of a indefinite noun or adjective [125].

Orthographic rules in Arabic allow for words to have alternative spellings [116, 117, 118]. because certain letter can be written in different ways, as the table shows.

The use of subordination is common in English sentence structure, but not so in

HAMZA or MADDA with ALEF	أ	آ	إ
HAMZA or MADDA is dropped	ا	ا	ا
Example	أمير	آمل	إحسان
Transliteration	Amir	Amul	AHsAn
Meaning	prince	hopeful	charity

Arabic. Co-ordination is more often used in Arabic and this usually makes subordination unnecessary [125]. Consequently, Arabic sentences tend to be great in length and they may even begin with "and" (*wa*, و). Table 4.9 shows a comparison of the two languages.

Owing to the flexibility of the word-root system, any Arabic body of text will probably have a greater variety of word types then a comparable body in English. Consequently, there may be Arabic words that have no direct equivalent in English. Unidentifiable words may also occur more often in unseen bodies of text.

4.4 Natural Language Processing for Arabic

Arabic poses two main difficulties for NLP; the addition of multiple prefixes and suffixes, and absence of words. A single word may be made up of many letters but contain only a couple of letters in its root. This makes grammatical classification and analysis very difficult for NLP and leads to ambiguity, especially for verbs [126].

Many researchers are aware of the problems of using NLP on Arabic and number of academic institutions and commercial concerns are addressing this, for example the Sakhr Alamiah computational linguistics group [121] has collected much of this work and has presented the state of Arabic linguistic analysis with regard to NLP as being in four stages:

1. Morphological analysis takes each word in isolation, regardless of its context. It then removes all prefixes and suffixes, leaving the word root or pattern.

Table 4.9: Comparison Between Arabic and English

Category	Feature	Arabic	English
Word Level	Derivational morphology	Complex, using roots and patterns (interdigitation)	Simple, using stems and affixes (concatinative)
	Inflection morphology	Highly inflected, systematic (concatinative)	Weakly inflected, less systematic
	Clitics	Usually equal to stop words	Many are reduced forms of words
	Capitalisation	None	Employed
Sentence Level	Writing direction	Right-to-Left	Left-to-Right
	Common word order	VSO	SVO
	Sentences structure	Favours coordination	Favours subordination
Resources	Computational resources	Under-resourced	Resourceful

this "string" of characters is then compared with a known database of words in various forms. This is a popular method as it does not involve having to analyse syntax [127, 128], and is therefore suitable for text mining.

2. Syntactic analysis focuses on how individual words fit into the context of a sentence and therefore requires a detailed knowledge of sentence and phrase structures.

3. Semantic analysis focuses on the meaning of words and how words are connected together. This method tends to remove ambiguities.

4. Statistical analysis is computationally expensive as it tries to classify each words in a sentence on the understanding that all sentences contain certain elements. It uses many techniques, including stemming, phrase grouping, and synonyms, and it recognizes the use of anaphora resolution, and roles of words.

4.5 Morphological Analysers for Arabic

A morpheme is a single linguistic unit that has meaning, and which cannot be subdivided. These include roots, prefixes and suffixes. Morphology is the study of these basic word-units and play a major role in the study of Arabic. Morphological analysis is central to NLP and many researchers have used it for information retrieval (IR), where clitics and suffixes are removed, and for machine translation (MT), where they are segmented, for example:

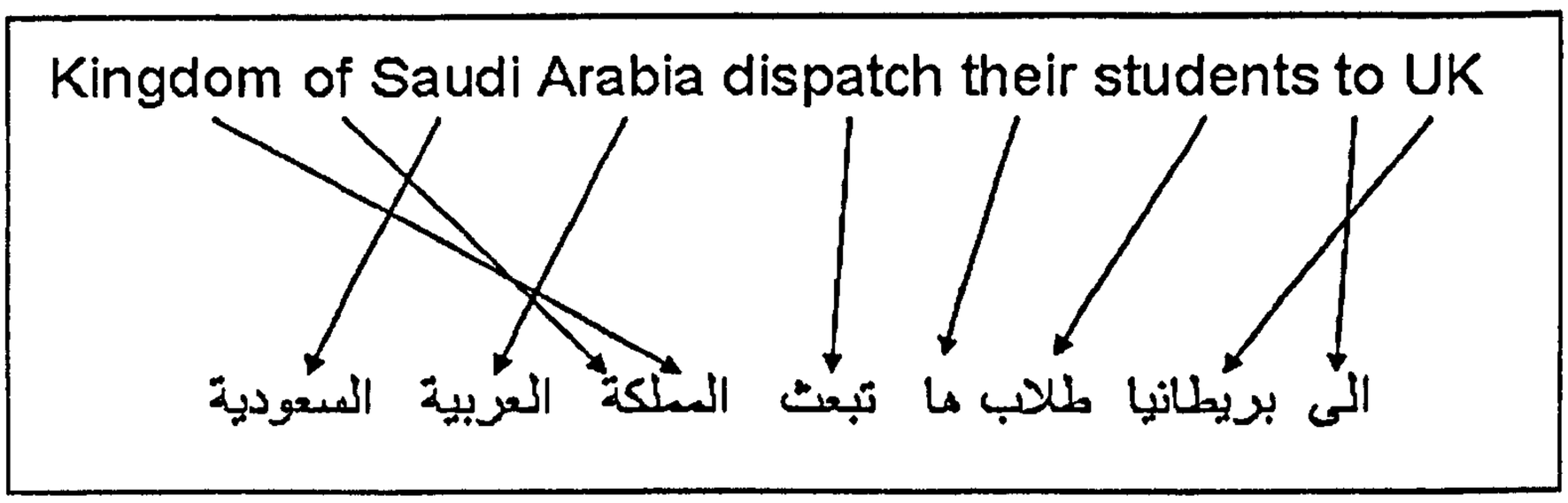


Figure 4.2: Alignment between morphologically analysed Arabic and English

Morphological analysis is particularly useful for Arabic as it determines the root and identifies prefixes and suffixes. All this information is classified and stored by a morphological analyser for information retrieval [129]. An Arabic morphological analyser has a pre-processing step in machine translation. Arabic morphological analysers can be classified into two classes: root-based ones that attempt to find the root of every analysable word by removing prefixes and suffixes, called "heavy stemmers" [130], and stem-based ones that peel only specific prefixes such as (و، ال، وَال، كَال، فَال) and suffixes such as (هَا، ي، ه ة، يه، ية) to find stems called "light stemmers" [129, 38, 131]. The affixes include inflectional markers for tense, gender, and number. Detecting affixes and clitics is not trivial because the same letters used for affixation can be part of a stem as well. Arabic morphological analysers are more prone to errors in domains where there are many proper nouns and borrowed words.

Whether researchers use stem-based or root-based analysis depends upon the researcher's goal. Recently, stem-based analysis has been faithful for IR, whereas root-based analysis has been shown to be more effective for in-depth studies of the language. [121] employed a stem-based technique for removing proclitics such as conjunction (*wa* و, *f* ف) and particles (*la* لـ, *be* بـ). However, this method required a great deal of training data, although not as great as for tokenization of strings

(tokens) separated by white-space.

Recent research has also shown that root-based analysers are not as precise as stem-based ones because roots can be part of words that belong to different classes. Semantics is an important part of pre-processing and of classification and so a morphological analyser must be as specific as possible. However, it seems that root-based analyses are too general and although stem-based ones also have a few difficulties (broken plurals can be misclassified), they are more precise.

To our knowledge, there are very few text mining studies on Arabic specialised texts in Arabic; there has been little academic research, and it has only focused on mining from monolingual texts.

4.5.1 Types of Arabic Morphological Analysers

The first stage in corpus analysis is having a morphology system for natural language processing. No application in this field can survive without a good morphology system to support it. This is particularly important for the Arabic language, which has its own attributes that are not found in other languages.

Arabic morphology is extremely complex with a large set of morphological features such as Part of Speech (POS), person, number, nominal case, determiner pro-clitic, en-clitic and nunation. These make Arabic very difficult to stem, and need both concatenative morphology (affixes and stem) and templatic morphology (root and patterns). That is why many researchers have worked in this area. Al-Fedaghi and Al-Anzi developed an algorithm to generate the root and pattern of a given Arabic word [132], and Khoja and Garside present an effective root extracting tool [133].

Therefore, with the Buckwalter stemmer it is possible to utilize Arabic morphology as the basis for Natural Language Processing (NLP). Furthermore, the algorithm used by [113] for the analysis of Arabic morphology also removes prefixes and suffixes

from Arabic words; this allows root-based algorithms to further reduce stems to roots.

4.5.2 Buckwalter Morphological Analyser

Tim Buckwalter wrote the Arabic Morphological analayser V1.0 (AraMorph.pl) and this is available on-line from Linguistic Data Consortium (LDC) [113]. Researchers may downloaded this for free and it includes its own algorithm. It consists of three lexical dictionaries for affixes, suffixes and stems and three sets of rules that specify how these lexicons can be overlapped to derive actual Arabic word tokens. The databases and the analyser work with Buckwalters transliteration system (See appendix A). The Arabic text has been transfered into Roman characters. This has been done by using XeroxBuckwalters transliterating system [134, 135, 113]. Moreover, this system enables each character from the Arabic set to correspond one-to-one of the Roman character set. One needs to bear in mind that this would not work on text containing both Arabic and Roman alphabets, which eventually saves work in ASCII code prior to analysis.

In this project, we adapted the Buckwalter stemmer to finding only roots, which are more difficult to discern than stems. The Buckwalter stemmer, in its original form gives all possible morphological analyses and diacritizations. It also produces transliterated Arabic with many possible meanings. The results of the Buckwalter stemmer can be viewed in Figure 4.3.

Our modified output provides only the single most reasonable Arabic root of a word (in Arabic script), as in the following example of Arabic text (roots):

قَاد جريح قَاد مركز أَمّ ال قَاد بلَاد فريق أَطَاح سنغَالي تبدّ زملَاءي

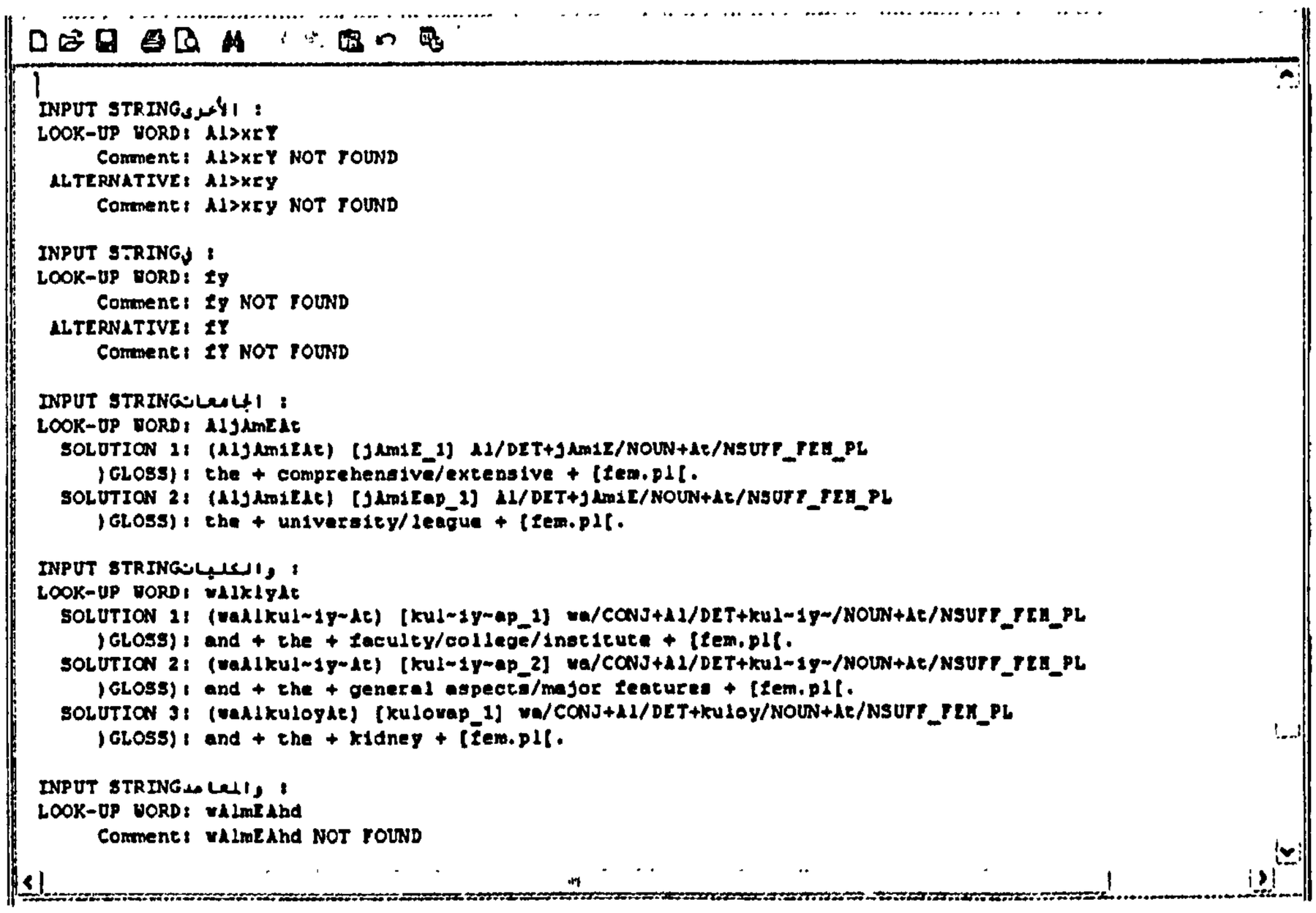


Figure 4.3: Screenshot transliterated Arabic

We know that roots are too abstract for effective information retrieval, and if the stemmer does not strip any affxes at all, then it is faulty. Although Buckwalter's stemmer made many mistakes, we extend on this previous work be experimenting with a wider range of pre-processing schemes for Arabic stemmenig in order to obtain better results.

In this reserch, we started to investegate hard to develop a stemmer concerning Arabic mainly. This effort has been made on the light of the work done by Buckwalter. Buckwalter [113] stemmer strips prefixes such as definite articles and conjunctions (و ، ال ، وَال ، كَال ، فَال) from the beginning words and suffixes (هَا ، ي ، ه ة ، يه ، ية) from the end of words, and provides better results than the khoja stemmer [133].

4.5.3 Sakhr's Morphological Analyser

In order to verify our procedure we need to compare our results with another system that is designed specifically for Arabic. Sakhr Morphological Analyser is such a

model and it provides morphological patterns after stripping affixes. This is called the Multi-Mode Morphological Processor (MMMP) [136]. Little work in NLP has been conducted with this system. Unfortunately, it has not been possible for us to activate the software and we have not yet been able to access the specifications that could help us.

4.5.4 Khoja and Garside Morphological Analyser

Khoja and Garside morphological analyser [133] is designed to peel away layers of prefixes and suffixes, then checks through a list of patterns and roots to determine whether the remainder could be a known root with a known pattern. If so, it returns the root, otherwise, it returns the original word–unmodified. This system also removes terms that are found on a list of 168 Arabic stop words. It is almost as effective as Buckwalter stemming, but tends to fail on foreign words, which are left unchanged thus it fails to remove definite articles and obvious affixes.

4.5.5 Comparison Buckwalter's vs. Khoja's Stemmer

Several experiments were conducted in order to evaluate the relative performance of Buckwalter's and Khoja's stemmer. Same set of documents were feed to both the stemmers. Results obtained are summarized in Table 4.10. It shows that Buckwalter's stemmer was 57% successful in finding the roots from the input documents while Khoja's stemmer fell short with 50%. In removing the stop words Buckwalker's stemmer again superseded with average of 42% success rate as compared to 34% of its counterpart. In a nutshell it is concluded that Buckwalker's stemmer is superior to Khoja's stemmer in almost all aspects.

Table 4.10: Comparison of Buckwalter Stemmer vs. Khoja Stemmer for Arabic Documents

	Buckwalter Stemmer				Khoja Stemmer			
	exp1	exp2	exp3	Correct	exp1	exp2	exp3	Correct
Documents	1	4	10		1	4	10	
Words	154	809	1751		154	809	1751	
Roots	89	410	1117	57%	76	357	1002	50%
Stop-Words &HTML	65	399	634	42%	52	519	346	34%
Normalization	√	√	√		×	×	×	

4.6 Affixation

The addition of prefixes, infixes and suffixes are central to the creation of meaningful words in Arabic. These affixes are important in the morphology and derivation of words in Arabic. According to [137], the purpose of morphology is "to segment words into their individual morphemes, prefixes, infixes and suffixes in order insight into the language". An example of this is the Arabic word for "the foods" (المأكولات, *AlmAkwlAt*), which can be broken down into its constituent parts as in Figure 4.4:

4.6.1 Multi-lingual Morphological Analysis (MMA)

We have created multi-lingual morphological analysis tool for Arabic-English languages, this model is easy to utilize when obtaining the roots of words and the indices of each root from the multi-lingual root dictionary. Our model has the ability to perform pre-processing stage for Arabic-English languages. Details are given in section 5.1.

4.6.2 Stem Method

This method requires some linguistic treatment for both documents and queries. Having said that, this method is deeply based on the word stem rather than the

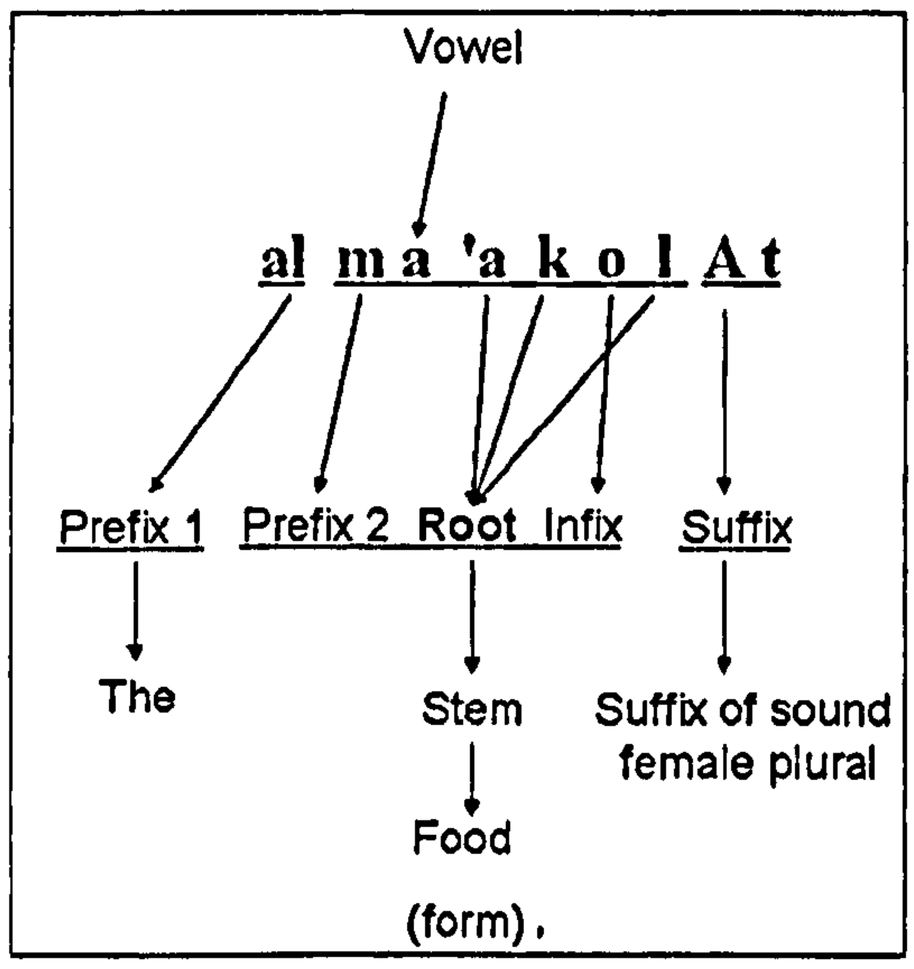


Figure 4.4: Word Segmentation

word itself. In other words, all prefixes and suffixes that may be attached to words are removed in order to unify words having the same stem under the same stem. This process may increase the success of matching documents to a query, but on the other hand, it could result into missing some relevant texts.

4.6.3 Root Method

The idea of this method is that words based on a common root should be reduced to that very root. Each term is then matched against the patterns to extract the letters corresponding to the standard word (fEala, فعل) of the tri-roots or (fElala, فعلل) of the quadri-roots, respectively. Roots with vowels or hamza, or two letter roots are also handled systematically. In other words, this method is able to retrieve all relevant texts of a query or keywords. The root based method performs well than the stem based, at all recall levels. Abu Salem [9] confirmed this efficiency of root

based method by re-performing the experiment first presented by Al-kharashi's [8], but now with different set of data [Abu Salem], [Al-kharashi]. The obtained results of Abu Salem are presented in Table 4.11 and Figure 4.5. Hmeidi also confirmed that the performance of root method is more effective than the words method [10].

Despite the limitations of stem method such as missing relevant texts, etc. it is still being used by current Arabic text retrieval systems. While the root method retrieval rate is much better than word or stem based methods, as can be seen from Table 4.11.

Table 4.11: Average Retrieval of 32 Queries of Titles [Abu Salam]

	Retrieved	Relevant	Irrelevant
Words	0.56%	0.56%	0.00
Stem	4.06%	3.84%	0.22
Root	6.69%	5.47%	1.22

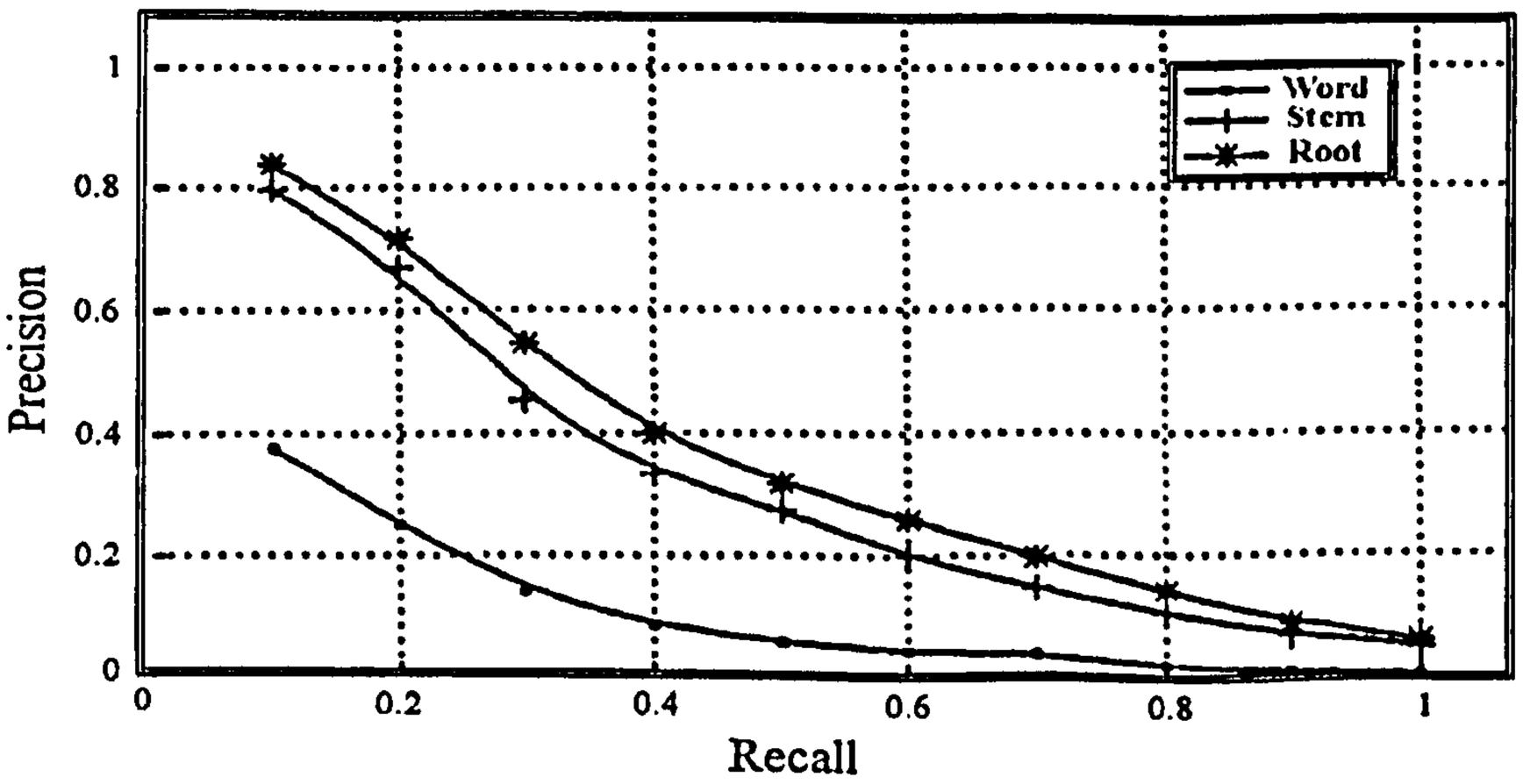


Figure 4.5: Average Recall-Precision plot for "Word", "Stem" and "Root" based methods, from [Abu Salam]

4.7 Summary

Arabic is a semitic language with many forms, but Modern Standard Arabic (MSA) is most widely recognized. Therefore MSA is used in language processing, and although not perfect, transliteration has been used in this regard. Usually, Arabic roots of words are made up of three consonants (sometimes four or five). Sentences are verb-subject-object but word order is not the main difficulty. Arabic morphology is extremely complex with the extensive use of affixes. Word roots give meaning with a sentences, and grammatical consistency with multiple prefixes and suffixes. A comparison of Arabic and English is presented in this chapter.

NLP on Arabic has difficulties because of the morphology which shows that an Arabic word may be long, but its root, after affix stripping may be very short. This can lead to ambiguities. However, computational linguistic analysis has been performed, and it has four main stages: morphological analysis, syntactical analysis, semantic analysis and statistical analysis. With regard to morphological analysis two approaches are reported in the literature viz: stem based and root based. Root based analysis has advantages over its counterpart as stated in Section 5.2.13. Thus in this study the root based analysis in conjunction with SOM is utilized. To the author's knowledge this combination is not previously used SOM for Arabic language. Modification to the Buckwalter root based morphological analyzer is made which successfully strips Arabic prefixes, suffixes and infixes.

We purposes that the Arabic stemmer algorithm can be applied on other languages such as Urdu and Persian. This is due to the similarities between the Arabic, Urdu and Persian languages. These languages use the same alphabet and calligraphy as that of Arabic. For example these words (حكيم,Hakim, wise),(طبيب, tabyb, doctor),(مسجد, masjid, mosque),(قَافلة, qAfilah, caravan),(كتَاب, kitAb, book),(مدرسة, madrash, school) have same spelling, same pronunciation and same meaning in three languages. Therefore, we have conclude that the model for Arabic can be

applied to both languages, Urdu and Parisian, the word "حکیم" is pronounced and transcript as the same in Arabic, with a slight modification on the bilingual dictionary composed by the researcher. The implementation of the morphological system is discussed in Chapter 6.

Chapter 5

SOM Approach For Multilingual Text Mining

Objectives

- To present the approach of MLTM for Arabic-English.
- To present Pre-Processing for Arabic-English.
- To present the Arabic stemming Algorithm.
- To present a general purpose of bilingual dictionary.
- To clarify why text classification is needed.
- To illustrate how to generate the indices a root.
- To present the Algorithm of SOMMLTM and its training.
- To present the Quality of Test and Data Visualization.

5.1 Introduction

This chapter deals with the prototype components such as mining knowledge and retrieval information for bilingual datasets, and describes the final model evaluated in this thesis. The first section briefly discusses the multilingual text mining model for knowledge discovery. The second section provides a methodology for the automatic combination of various lexical resources such as a multi-lingual dictionary, while the third section provides the algorithm for multilingual text mining. Finally, data visualization using SOM is demonstrated. The overall processes of the framework and its components are illustrated in Figure 5.1. The framework consists of 4 stages:

- Stage 1: Pre-processing and indexing components.
- Stage 2: Training of SOM, and generation of clusters.
- Stage 3: Quality of test.
- Stage 4: Graphical user interface.

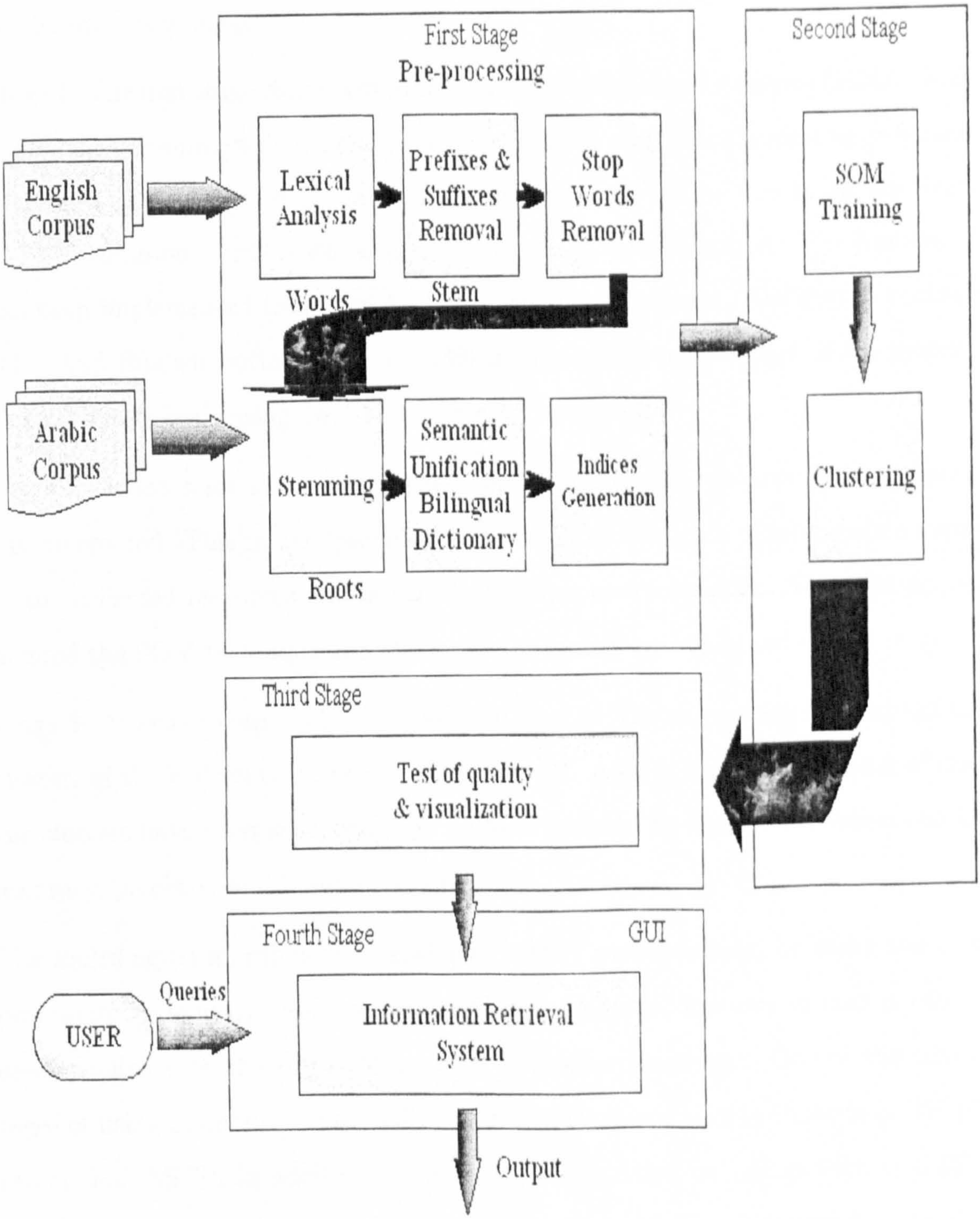


Figure 5.1: Framework for Multilingual Text Mining

Figure 5.1 shows the General Framework of Multilingual Text Mining. Our approach has the capability to perform four stages. In the next section we will concentrate on the first three stages.

Stage 1: The first stage consists of Multilingual Morphological Analysis (MMA),which is language pre-processing, and involves stemming and lexical semantic resources such as a multilingual dictionary. Most of the problems in this stage are likely to be uncommon words, such as named entities or transliteration. The framework has been implemented using the Arabic Morphology program (AraMorph) package [113], and English porter Stemmer [138] for the pre-processor part of the system. The system is built using Java Programming Language.

Stage 2: Once a set of resources has been obtained, the resource feature matrix can be created. This matrix provides an overview of the types of information found in the collected resources and of certain resource characteristics. Then, we implemented the SOM technique and the results collected are visualized on the map.

Stage 3: After the map of clusters has been created, the correctness (quality) of the clusters of the various text needs to be evaluated. After it has been established that the clusters have been successful, the SOM is assessed by numerical evaluations for accuracy, precision and human classification.

The multilingual morphological analysis (MMA) was developed by using the Java programming language and was designed with universal features so that it can be easily modefied in the future to go well with other languages. One of the advantages of using Java is that internally, programs represent text in Unicode (UTF-16) rather than ASCII. In addition, it gives support for other encodings such as UTF-8 and ASCII. Its graphical user interface components were also designed to be able to display right-to-left languages. Since Java provided many services to process and show Arabic. It works for other languages such as Hebrew, Japanese and many other languages. The other benefit of Java programs is that it is multiplatform. As

long as the Java Runtime Environment (JRE) is installed on the system, it can be run a Java application [11]. As a result, we have been used it to generate our own system for Multilingual Morphological Analayser.

We have divide our model into four main stages, brief descriptions on the first three stages are as follows.

5.2 Stage I (Pre-processing)

This stage has been implemented using Arabic Morphology program (AraMorph) Buckwalter stemmer [113] and English porter Stemmer (Stemmer) [138]. Quit modification in the original Arabic Morphology stemmer was also done. Therefore, the original Arabic stemmer reads the words but does not remove the non-alphabetic words, stop-words and html tags. The stemmer then gives all possible morphological analyses and diacritizations. It also produces transliterated Arabic with many possible meaning and transliterates into Roman characters, which are more difficult to discern than stems. Our adaptations for this purpose involved several modifications of the stemmer process, modifications that appear to be helpful in this sense. Our modifications for the Arabic stemmer are now as follows:

- Read the all documents.
- Remove the stop-words and html tags.
- Remove the prefix and suffix to obtain the roots
- Remove all words less than three characters.
- Remove diacritics.
- Replace أ, إ, and آ with ا.
- Replace final ي with ى.

- Replace final õ with o.

The outputs of the original Arabic morphology were Arabic words in English alphabets with a long chain of possible English meanings but our modified output is only the single most plausible Arabic roots of the words for whole corpus and saved in ASCII codes . Also, the English Stemmer was capable of reading only a single English word and stemming that. We modified the structure of the code to read the whole corpus, to eliminate stop-words from each document and remove the prefixes from the words and to generate the root of each document (see the Java code in Appendix D). The results are a pure single text "read.txt" file and saved in ASCII codes. The execution process of AMESD which compares the "read.txt" file which composed all roots from both languages with bilingual dictionary. Finally the table of indices is generated by AMESD after the comparison process. For a more detailed description of the stemmer see section 5.1.7. The requirements of this stage will drive the second stage.

In the stage of pre-processing the available texts from both the languages are read. These are then lexically analysed and are resolved into words. These words are then passed through the filtering process i.e. eliminating stop words such as articles, conjunctions, prepositions, punctuation, auxiliary verbs etc. (examples are: he, is, and, in, the, !, .etc).

5.2.1 Pre-Processing Multilingual Analyser

We have created a multilingual pre-processing analyser for Arabic-English, and this model is easily utilized to derive the roots of words and the indices of each root from the multilingual root dictionary. Figure 5.2 shows the screen shot for the framework of the multi-lingual morphological analysis below. Our model has the capability to perform pre-processing stage.

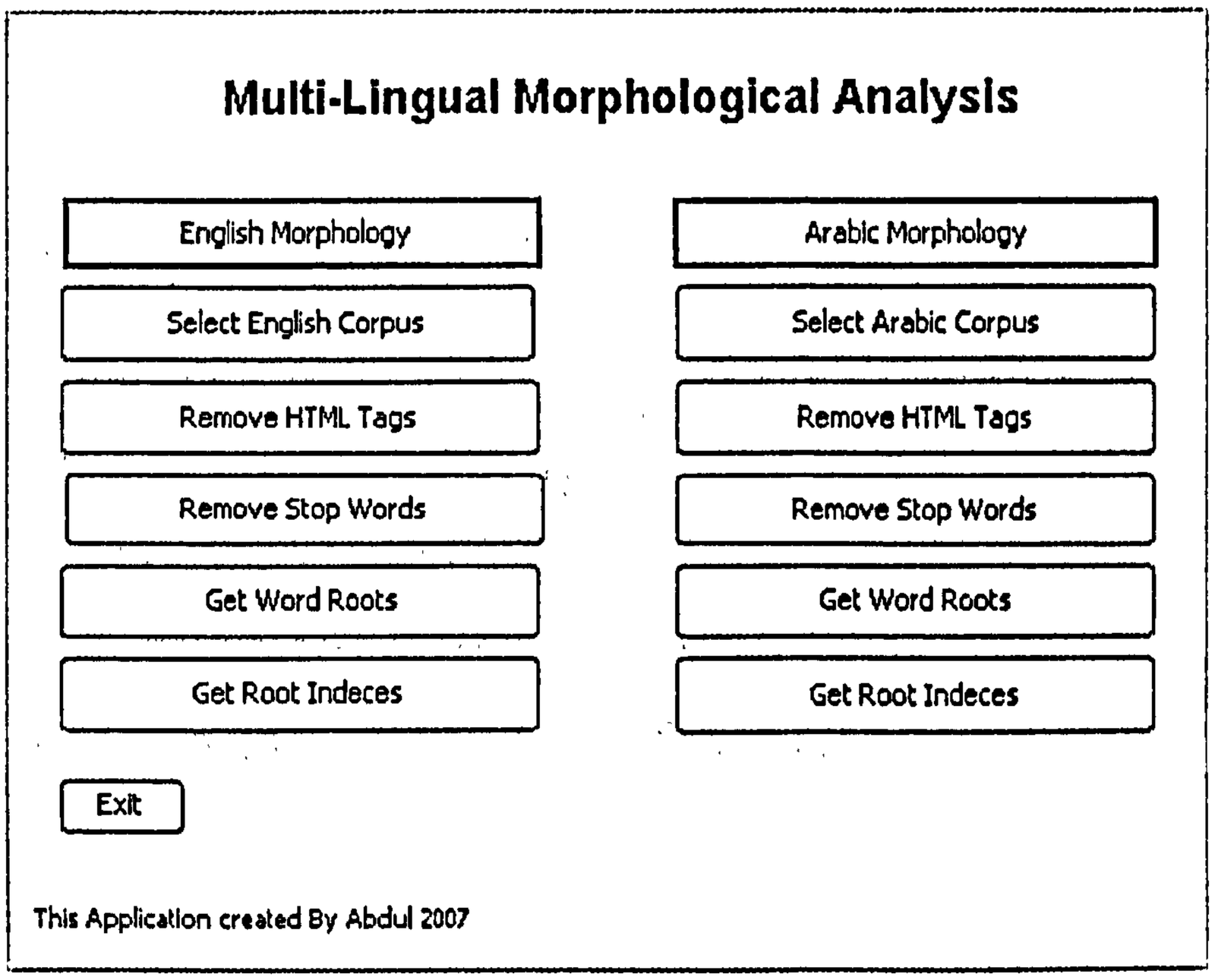


Figure 5.2: The Model of Pre-Processing Stage for Arabic-English

5.2.2 Lexical Analysis

Natural language process systems need a lexical analysis that breaks the text down into the smallest units with white space into a sequence of characters known as tokens [139]. A token can be words, symbols, html tags, .etc; therefore, it will be a useful part of the structured text. From the lexical analysis point of view, Arabic is a complex and nature language. The input corpus of text documents for Arabic undergo lexical scale through these operations that include:

1. Checking the token = Arabic or English.

2. Checking the token = stop words.

3. Changing the letter before the stress into a duplicate.

4. Remove all diacritics.

5. Changing the expanded alif آ to alif ا.

6. Deleting punctuation, numerals, HTML tags and not Arabic codes.

7. Deleting repeating spaces and repeated tabs.

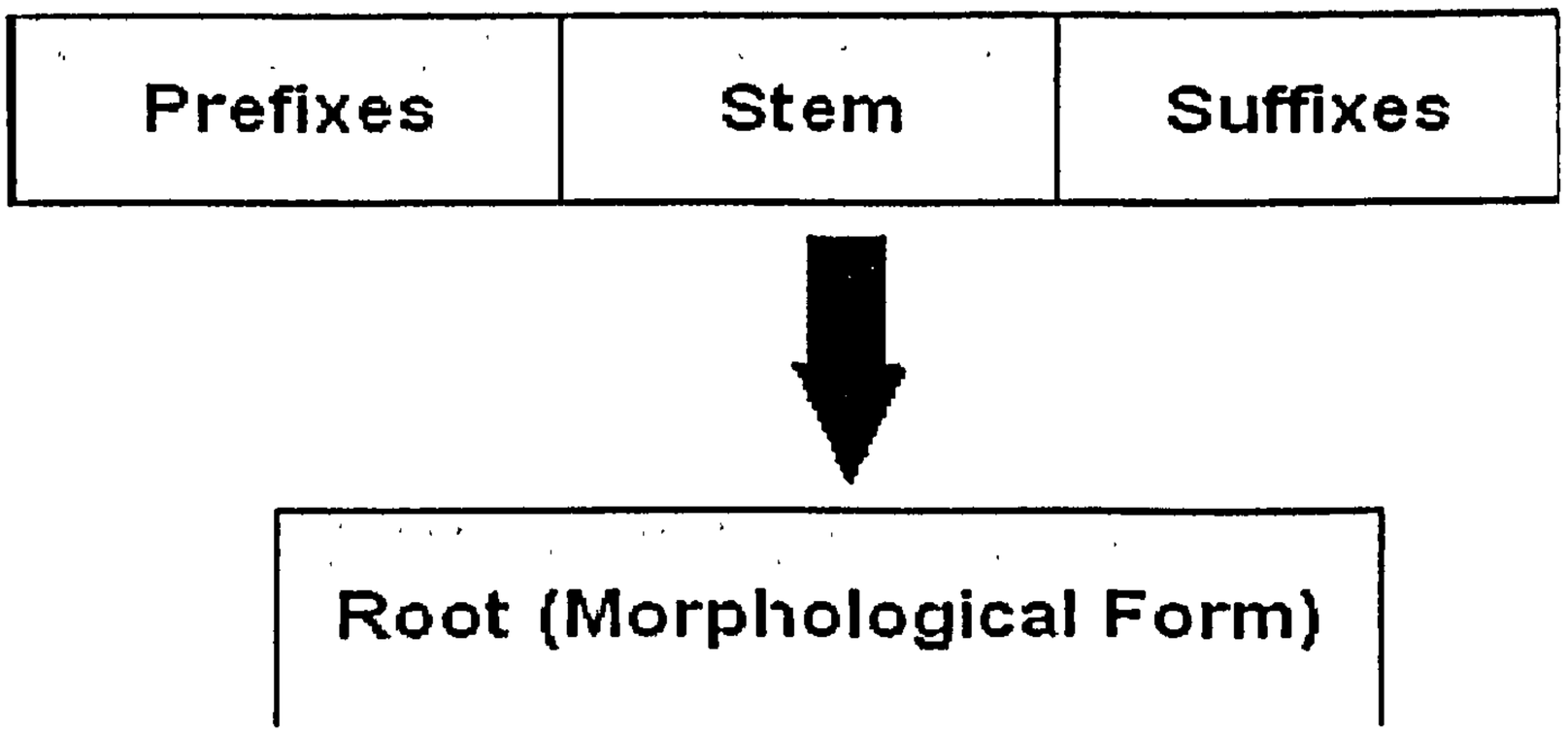
We have not finalized the complex lexical analysis of multilingual corpus; this is left for future studies.

5.2.3 General Stop Word List

In this component of this stage, we consider Arabic-English stop words, which are uninterrupted sequences composed of letters (a....z), digits(0...9), or two special characters (@ and _) of each language. We have defined a general stop word list for those words that serve no purpose for retrieval, but are used very frequently in composing documents. By filtering out such words, the message becomes clearer or more useful and this stop word list is developed for two reasons: firstly, we hope that each match between a query and a document will be based on good indexing terms. Thus, retrieving a document because it contains words like "that", "he", "when", "الذي", "هو" and "متَى" " in the corresponding request does not to form an intelligent search strategy. These non-significant words represent noise, and may actually damage the retrieval performance because these words do not distinguish between relevant and non-relevant documents. Secondly, we expect to reduce the size of the inverted file, hopefully in the range of 30 to 50%. The stopwords such a large number of articles, conjunctions, preposition, punctuation, auxiliary verbs and pronouns etc. (examples are: he, is, and, in, the, !, .etc). For instent, the SMART information retrieval system at Cornell University has 571 English words in its stop-word list; Fox [140] describes the final product of stop-word as 421 words in English Appendix B includes the list of English list. Arabic stop-word removal was performed after morphological normalization using a 168 stop-word list from the universities at Lancaster [118, 38]. See the list of Arabic stop-words in appendix B.

5.2.4 Prefixes and Suffixes Removal (PSR)

This process is based on the inflectional morphology of Arabic, and the main function of this process is to remove prefixes and suffixes attached to words. To do this process, two lists of prefixes and suffixes are created. The following decomposition is then presented, and finally the attached prefixes and suffixes are removed. The result of this process is a stem only. The stem consists of two parts not separated from each other; root (الجذر) and the morphological forms (الأوزان). A further description of this process can be seen from the example below:



The following example presents how the PSR remove the attached prefixes from the word library, "المكتبة". First, it will remove the attached prefixes "ال " from the word, since there is no suffix in this word the code will not do any thing to the word. After the process is done to this word, the result will be "مكتبة" as the stem. Finally, the stem will pass to the next process morphological analyser to find the root of the word "كتب" and the morphological form is " فعل " (composed form).

5.2.5 Arabic Root-Based Algorithm

All Root-Based stemmers have the same technique for finding the root of an Arabic word. Generally, they reveal the root of the word after stripping the prefixes and suffixes attached to the given word. The Root-Based stemming algorithm is illustrated below. This algorithm works on Arabic texts, taking the following steps per

word in the text:

Check the word w=($w_1, w_2, ..., w_n$) in the text.

1. IF w != Arabic THEN remove it.

2. IF not w THEN remove it.

3. Remove orthography with diacritics.

4. Remove prefixes.

5. Remove suffixes.

6. IF $w < 3$ letters THEN remove it.

7. IF $w > 3$ letters THEN remove the infix.

8. Check the w with morphological form.

9. Normalized the word.

Firstly, the algorithm checks the word is an Arabic word, and if it is less than 3 letters or is not a word, it is unimportant. Then it removes the orthography with diacritics, followed by prefixes and suffixes. It then checks if the word is greater than 3 letters and removes the infix. It then examines a list of morphological forms and roots to determine whether the remainder could be a known root with a known form applied. If so, it returns the root. Otherwise, it returns the original word, unmodified. Finally, it normalizes the word as presented below.

Often, vowel replacement and letter omission are required to construct words. There are different approaches to Arabic stemming e.g. algorithmic light stemmers which remove prefixes and suffixes [118]. Since morphological analyses in Arabic results from the addition of prefixes and suffixes as well as infixes, simple removal of suffixes is not as effective for Arabic as it is for English, which attempts to find roots, and

statistical stemmers, grouping word variants using clustering techniques. The corpus were normalized according to the following steps [117, 133]: .

- Replace أ , إ , and آ with ا.
- Replace final ي with ى.
- Replace final ة with ه.

5.2.6 Stemming In Arabic

Arabic language is composed of approximately 10,000 roots [141]. The words are formed from these roots with combination of prefixes, suffixes or infixes. Roots usually consist of three letters, four letters, or rarely five letters for example.

Suffix	شجـر	⟸	شجـرة
infix	نظـم	⟸	نظـام
Prefix	كتـب	⟸	يكتب

5.2.7 Stemming In English

In the case of English text, the documents are read and passed through the filtering processes i.e. eliminating stop words. When defining a stemming algorithm, the first process will only remove inflectional suffixes and past tenses and then remove the derivational suffixes. For example, Porter's stemmer [138], Lovins's stemmer [142] is based on a list of over 260 suffixes, KSTEM stemmer is heavily dependent on the entries in the dictionary being used [143], and The Paice/Husk Stemmer is used just one table of rules [144]. Moreover, after removing the stop words, an

indexing procedure tries to conflate word variants into the same stem or root using a stemming algorithm. When applying the stemming algorithm, first step builds the basic forms of the words (root) by removing the suffix "s" from plural word forms, removing the "ed" which is the past particle ending, and "ing" of the gerund ending from verbs, for example:

$$(suffix: \ cars \ \Rightarrow \ car \ , \ playing \ \Rightarrow \ play \ , \ walked \ \Rightarrow \ walk \ )$$

More complicated stemming for English words have also been proposed for the removal of derivational suffixes such as "-tion", "-ize", and "-hood" for example:

$$(suffix: \ action \ \Rightarrow \ act \ , \ generalize \ \Rightarrow \ general \ , \ childhood \ \Rightarrow \ child \ ,)$$

Second, the algorithm will remove the prefixes "en-", the "pre-" and "dis-" from the beginning of the verbs, for example:

$$(prefix: \ en-code \ \Rightarrow \ code, \ pre-process \ \Rightarrow \ process \ , \ dis-appear \ \Rightarrow \ appear \ )$$

5.2.8 Porter Stemmer

The stripping of suffixes from words has been widely investigated [138], according to the literature, and a variety of strategies have been employed. These strategies depend upon the purpose of the task and whether or not dictionaries (stemming dictionary or suffix list) are being utilized. Should the aim be improving the performance of information retrieval, then a suffix list will be sufficient although commands will also be needed to control the stripping of a suffix so that a meaningful stem remains.

The Porter Stemmer suffix stripping algorithm was introduced in 1980, together with rules, and thereby offered an "automatic means" for improving information

retrieval. However, it is most usefully employed when one extracts small vocabulary lists from large text files rather than applying it to a large amount of continuous text.

In the English alphabet, there are 26 consonant letters, including 5 vowels (a, e, i, o, and u) and the letter y. The letter y can be either a vowel or a consonant; in 'story' it is a vowel, in 'stay' it is a consonant.

This will be represented by the single form (upper case):

$$[C]VCVC...[V]$$

Below, the form indicates that the (VC) is repeated m times, where m is called the measure of any word or part of word when represented in this form:

$$[C](VC)m[V]$$

The "condition" part may also contain the following:

*S - the stem ends with S (and similarly for the other letters).

v - the stem contains a vowel.

*d - the stem ends with a double consonant (e.g. -TT, -SS).

*o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

Step 1a

SSES – > SS	caresses – > caress
IES – > I	ponies – > poni
ties – > ti	
SS – > SS	caress – > caress
S – > cats – > cat	

Step 1b

(m>0) EED – > EE fee – > feed

agreed – > agree

(*v*) ED – > plastered – > plaster

bled – > bled

(*v*) ING – > motoring – > motor

sing – > sing

If the second or third of the rules in Step 1b is successful, the following is done:

AT – > ATE conflat(ed)– > conflate

BL – > BLE troubl(ed)– > trouble

IZ – > IZE siz(ed) – > size

(*d and not (*L or *S or *Z))

– > single letter

hopp(ing) – > hop

tann(ed) – > tan

fall(ing) – > fall

hiss(ing) – > hiss

fizz(ed) – > fizz

(m=1 and *o) – > E fail(ing) – > fail

fil(ing) – > file

In order to map a single letter, the rule is to remove one of the double letter pair. The -E is put back on -AT, -BL and -IZ, so that the suffixes -ATE, -BLE and -IZE can be recognised later. This E may be removed in step 4.

Step 1c

(*v*) Y – > I happy – > happi

sky – > sky

Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

Step 2

(m>0) ATIONAL	– > ATE	relational – > relate
(m>0) TIONAL	– > TION	conditional – > condition
rational	– > rational	
(m>0) ENCI	– > ENCE	valenci – > valence
(m>0) ANCI	– > ANCE	hesitanci – > hesitance
(m>0) IZER	– > IZE	digitizer – > digitize
(m>0) ABLI	– > ABLE	conformabli – > conformable
(m>0) ALLI	– > AL	radicalli – > radical
(m>0) ENTLI	– > ENT	differentli – > different
(m>0) ELI	– > E	vileli – > vile
(m>0) OUSLI	– > OUS	analogousli – > analogous
(m>0) IZATION	– > IZE	vietnamization – > vietnamize
(m>0) ATION	– > ATE	predication – > predicate
(m>0) ATOR	– > ATE	operator – > operate
(m>0) ALISM	– > AL	feudalism – > feudal
(m>0) IVENESS	– > IVE	decisiveness – > decisive
(m>0) FULNESS	– > FUL	hopefulness – > hopeful
(m>0) OUSNESS	– > OUS	callousness – > callous
(m>0) ALITI	– > AL	formaliti – > formal
(m>0) IVITI	– > IVE	sensitiviti – > sensitive
(m>0) BILITI	– > BLE	sensibiliti – > sensible

A program switch is needed to test the string S1, which can also be made faster depending on the switch that tests the penultimate letter. This process will fairly determine the possible breakdown values of the string S1. It will be seen in fact

that the S1-strings in step 2 are presented here in the alphabetical order of their penultimate letter. Similar techniques may be applied in the other steps.

Step 3

(m>0) ICATE – > IC	triplicate – > triplic
(m>0) ATIVE – >	formative – > form
(m>0) ALIZE – > AL	formalize – > formal
(m>0) ICITI – > IC	electriciti – > electric
(m>0) ICAL – > IC	electrical – > electric
(m>0) FUL – >	hopeful – > hope
(m>0) NESS – >	goodness – > good

Step 4

(m>1) AL – >	revival – > reviv
(m>1) ANCE – >	allowance – > allow
(m>1) ENCE – >	inference – > infer
(m>1) ER – >	airliner – > airlin
(m>1) IC – >	gyroscopic– > gyroscop
(m>1) ABLE – >	adjustable – > adjust
(m>1) IBLE – >	defensible – > defens
(m>1) ANT – >	irritant – > irrit
(m>1) EMENT – >	replacement – > replac
(m>1) MENT – >	adjustment – > adjust
(m>1) ENT – >	dependent – > depend
(m>1 and (*S or *T)) ION – >	adoption – > adopt
(m>1) OU – >	homologou – > homolog
(m>1) ISM – >	communism – > commun
(m>1) ATE – >	activate – > activ
(m>1) ITI – >	angulariti – > angular

(m>1) OUS – > homologous – > homolog

(m>1) IVE – > effective – > effect

(m>1) IZE – > bowdlerize – > bowdler

The suffixes are now removed. All that remains is a little tidying up.

Step 5a

(m>1) E – > probate – > probat

rate – > rate

(m=1 and not *o) E – > cease – > ceas

Step 5b

(m > 1 and *d and *L) – > single letter

controll – > control

roll – > roll

When the stem is too short, the algorithm is careful not to remove a suffix. The length of the stem is being given by its measure, m. We should note that there is no linguistic basis for this approach.

5.2.9 Lovins Stemmer

Julie Beth Lovins first presented her stemming algorithm [142] in 1968, and it is a single-pass, context-sensitive and longest-match stemmer. Due to its nature of being a single pass algorithm, removes a maximum of one suffix from a word. It exploits a large list (297) of endings, each of which is associated with one of a number of qualitative contextual restrictions that prevent the removal of endings in certain circumstances. Thus this is a context sensitive stemmer that removes endings in a single pass, based on the longest-match principle. Although it employs certain rules in order to deal with the most common exceptions, all endings are subjected to a default exception. This is designed to prevent the production of ambiguous stems,

and states that any stem must constitute at least two letters. The Lovins stemmer is usually a little more cautious than Porter's but nevertheless can suffer from over non-word stems. Three more rules are imposed before the removal of an ending, and they are:

- Increasing the minimum length of a stem following a ending's removal.
- Preventing the removal of endings when certain letters are present in the remaining stem.
- Combinations of the above restrictions.

Unfortunately, researchers have been unable to identify many examples that exactly fit the Lovins rules because there are usually exceptions for endings, and this results in erroneous stems. Unfortunately such exceptions are usually particular and specifically associated with an ending, and thus a disproportionately large number of rules would have to be designed just in order to prevent a small number of errors. This would cost a great deal in terms of time and data, and improvements in performance would probably become negligible with repeated iterations. It is therefore usually the case that common exceptions are incorporated into the list and a number of errors, hopefully small, are ignored.

Stems sometimes have alternative spellings and it these that are referred to as 'spelling exception'. In English, most of these are due to Latinate derivations such as matrix and matrices but researchers have isolated other types of exceptions, such as differences in British and American spellings (through and thru), and the doubling of consonants in order to maintain inflexion when a suffix is added (begin and beginning). Lovins herself was aware of these difficulties and proffered two solutions (recoding and partial matching), and therefore a recoding phase is included in the algorithm but for more details on her reasoning, please refer to her original paper.

5.2.10 KSTEM Stemmer

The KSTEM stemmer introduced a new approach to stemming, based on machine-readable dictionaries and well-defined rules for inflectional and derivational morphology. Although this stemmer addresses many of the problems associated with both Porter and Lovins, its overall performance is not consistently better it is heavily dependent on the way that data is entered. KSTEM checks the current string against the dictionary before removing a suffix but a dictionary usually lists word forms separately if they have different meanings and does not list them as in a thesaurus. For example, 'journal' and 'magazine' have differing word forms but similar meanings, and this needs to be taken into account [145, 143].

5.2.11 Paice/Husk Stemmer

Chris Paice and Gareth Husk first published their conation-based iterative stemmer in 1990, and called it the Paice/Husk [144] stemmer. They controlled the removal and replacement of endings by developing a table of rules that is sub-divided into sections. Each section incorporates a number of rules that specifically accord to the final letter of a suffix, making this a time-efficient method of identifying a word or truncated word. A section of the rules may also specifically accord to 'intact' words, where no recoding or partial matching is necessary. Paice and Husk have thereby developed an efficacious algorithm that identifies targets quickly and stems them effectively. effective.

5.2.12 Semantic Unification of Bilingual Dictionary

The dictionary created does not include dictionary-inflected forms, but does include root forms. These correspond to the root and some nouns. The total number in our bilingual dictionary is over 6,000. The reason for the creation of a root dictionary is to be able to retrieve all relevant texts of a query or keywords in both languages.

Furthermore, we have used text classification in our dictionary, and in order to do so, the roots in the bilingual dictionary are encoded in English number-Arabic number (NENA) structure.

5.2.13 Dictionary Encoded

1. We use roots instead of words because every language has its own grammatical structure, and to avoid this problem, employing "roots" is the best option. It also reduces the search time in dictionary.

2. Each root assigned NENA a unique number.

3. NE and NA stands for 4 digits English and 4 digits Arabic indices.

4. We start from the first word appearing in the English to Arabic Dictionary "accept" meaning in Arabic "قبل".

5. 1001 is assigned to "accept" and 0001 to "قبل".

6. For every new word either in English or Arabic a new index is assigned.

7. If the word is repeated due to its multiple meaning, then the index remains the same.

8. Finally, the number of English and the number of Arabic are combined together.

There are three main advantages to selecting the "root approach" and they are given below. This is supported by the outcome of this study supported by Al-kharashi [8], Abu salem [9] and Hmeidi [10].

1. It is able to retrieve all relevant texts for a user query.

2. It reduces the search time in the dictionary.

3. It reduces the memory spaces.

5.2.14 Text Classification

While the encoding procedure is under process, it is hard to decide on which text category an item belongs to and in which domain. [146], examines in detail the problems of text classification and reports that corpus design makes use of some internal and external criteria to decide on the text category, and sometimes they are of quite a different order when compared with external classifications. Many internal and external criteria reflect each other. Sinclair proposed that many text classifications are noticeably based on topics as they are represented in newspapers and magazines. Although he thinks that this is "a valuable feature of reflexivity of language", he says that "a typology based on such criteria will be untidy". One of the common internal criteria is the choice of vocabulary in a certain text, and as a result he proposed 35 categories. Sharoff [147] has presented and recommended another type of classification, because the previous one is "too fine-grained". The new classification consists of only 8 main categories or general domains that include other types of text. Table 5.1 shows general domain classifications of the text types depending on [147]:

Table 5.1: Text Types Classified in Main Domains

Code	Domains	Keyword	Text types
1.	Natural Science	Ns	math, biological, physics, medical, etc.
2.	Apply Science	As	agriculture, medicine, ecology, engineering, computing, etc.
3.	Social Science	Ss	law, history, philosophy, language, education, etc.
4.	Politics	Po	inner, world
5.	Commerce	co	finance, industry
6.	Life	Li	fiction, conversation, advertisements, rest, menus etc.
7.	Arts	Ar	visual literature, architecture, performing
8.	Leisure	Le	sports, travel, entertainment, fashion, etc.

Some of the text types in the above table can be classified under several domains depending on the topic that they handle. For example, "interviews" can handle general topics but they can handle more specialised topics such as politics or medicine as well. Also autobiography, memos, and patents can be applied as text types.

Alongside collecting the texts, we created a bilingual dictionary in Microsoft Access which stores the ID-Word, English-Word, Arabic-Word, Cat-No of words in a dictionary. This dictionary is important for having the label of each text of the input corpus.

In this study, we have chosen the Sharoff text classification method because of its clarity and plausibility. Methods like lack these qualities; "Text Encoding Initiative" (TEI). According to Sinclair "a typology based on such criteria will be untidy" [146].

5.2.15 Indices Generation

In order to check the functionality of Multilingual Pre-Processing Analyser, we selected three Arabic and four English documents [148] as the test case. The Multilingual Pre-Processing Analyser was then executed to derive the final result i.e. the table of indices. We began with documents in both English and Arabic.

The first document in (English) is related to education, the second (Arabic) document is the exact translation of the first document. The third (English) document discusses activities. The fourth document talks about education (in Arabic). The fifth document in English discusses fast food and Americans' attitudes. The sixth document (Arabic) is related to sport, and finally the last document in English is the exact translation of the seventh document.

Finally, the output is a single document containing the roots from both sets of documents (Arabic-English). We expect that after using this set of documents the

output of the documents related to sport will have the most indices in common, as will the documents about education. The indices generated by MMA, for the set of seven documents mentioned above. The first, third, fifth and seventh columns correspond to the four documents in the English language and the second, fourth and sixth columns correspond to the Arabic documents. The table is generated by assigning index number to words occurring in the two sets of documents from the Arabic-English dictionary.

The results achieved by MMA of the two datasets consist of seven columns and twenty-seven rows of relevant words in both languages. Each column corresponds to a document in either Arabic or English. Hence the first row and first column is "10340054" in English related to "student", while "10340054" from the first row and second column is presented to "طلب" in Arabic, then the first row and third column is "10330053" indicated to "sport" in English, while "10290047" is related to "قدم" from the first row and fourth column. Next the first row and fifth column "10030003" represents "امريكي" in Arabic, while "10640094" in the first row and sixth column represents "زاءير" in Arabic, and the first row and seventh column is "10410070" related to "black" in English. So we expect some common indices in the Table 6.3.

5.3 Automatic Mining of Documents

For the mining of documents from a corpus of special languages texts, it is important to focus on the keywords. The documents or common sentence structures in which the keywords are embedded are assumed to comprise the principal elements of a subject specific to the document, which may help in mining text.

The way in which we approached the text mining and the algorithm are presented below. Given a specialist corpora (A_L), (E_L), terms are identified by a lexical

analysis, then the stemming systems to find roots of words for both languages are used. Next, we construct a vector (A_{Nword}) consisting of all unique words, and then construct a matrix (WC_{doc}) to compute the frequency of the terms in the dataset. We then replace the labels and the categories for all documents. After that, we compute the minimum distances for all units j and match the best matching unit (BMU). Finally, we update the weights of the BMU and its neighbourhood. Below is a demonstration of how the algorithm works using Arabic and English texts.

5.4 SOM for Multilingual Text Mining Algorithm (SOMMLTMA)

To present the SOMMLTM algorithm in its entirety we need to follow the steps below. The algorithm consists of three parts and these are: analysis, structure, and visualization. In the first part, the algorithm analyses each document in the corpus by using the lexical analysis, which then reads all tokens in both languages. Afterwards, we apply the pre-processing stage to enable filtering the corpus of stop words and html tags. After that, the morphological analysis strips the suffixes and prefixes from the stem to generate the roots. Finally, the system assigns the indices for all roots.

1. Analysis bilingual corpus A_L, E_L as input.

   - Lexical analyses for both languages as a token:

     $A_L = (f^1_{a1-Nword}, f^2_{a1-Nworg}, \cdots, f^{Ndoc}_{a1-Nword})$

     $E_L = (f^1_{e1-Nword}, f^2_{e1-Nword}, \cdots, f^{Ndoc}_{e1-Nword})$

   - Full linguistic analysis, combination of orthographic normalization, prefixes and suffixes removal.

   - Assigned each root by indices.

- Combining the corpus into AE_L matrix.
- $AE_L = A_L + E_L$

2. Examine the language of each document (if $AE_L = 1$, the document is Arabic else it is English).

3. Construct a matrix Doc_{all} containing 1...$Nword$ and 1...$Ndoc$, to make all columns identical in length, filling all columns to 10000000 that consist of less length. The words are encoded numerically.

4. Construct a vector A_N, N number of rows containing all unique words from each document.

5. Construct a matrix WC_{doc} containing 1...$Nword$, and 1...$Ndoc$.

6. For k^{th} document $1 <= k <= Ndoc$

$Doc_{all} = (x^1_{1-Nword}, x^2_{2-Nword}, \cdots, x^k_{3-Nword}, \cdots, x^N_{i-Nword} doc)$

for j=1:Ndoc

for k=1:N

$index\_Doc_{all}(k) = k;$

$word = Doc_{all}(i, j);$

if $word == A_{Nword}(k, 1)$

$WC_{doc}(k, j) = WC_{doc}(k, j) + 1;$

break;

end;

end;

end;

end;

where WC_{doc} calculates the frequency of each word per document.

Second part: the algorithm will prepares the datasets by adding the label[1] and category [2] for each vector in the matrix WC_{doc}, which will input datasets to the third part of the algorithm.

7. For each vector x_i^k in WC_{doc}, do $9-10$

8. Add a lable to each word

9. Add a lable to each vector x_i^k

10. Prepare the structure bilingual datasets ASMA, (snippet from main program).

$c\_f = 0;$

for $k = 1 : g\_d$

if $c\_n(i) == g\_d(k, 1)$

$c\_f = 1;$

if $L\_ind(i)==1$

$c\_n(i) = Cat(g\_d(k, 2), 2);$

elseif

$c\_n(i) = Cat(g\_d(k, 2), 1);$

end;

fprintf(fid,'%s',c_n(i),'_n');

break;

end;

end;

where the c_f is a variable for category-found, g_d is get from dictionary and c_n is category name.

[1] Label: A descriptive text in a cell that is not used for any calculations.
[2]Category: is a group of any sort of similar items.

Third part: the training and visualizing the dataset ASMA. This gives possible values of the results, which at a later stage can be seen as the presented results on the grid.

11. Visualize the output on the grid.

Pseudocode of learning mechanism from the SOM algorithm

```
loop forever
        Initial the parameters radius, learning rate and weight vectors.
        Select input data sets.
        Calculating the Euclidean distance.
        Find the BMU node.
        Training (the updating the weight matrix).
        Repeat until reached the criterion.
Endloop
```

Pseudocode of evaluation

```
loop forNword
   loop forNdoc
      Calculating the Euclidean distance of each node from BMU.
            Sum over all the occurring words.
            The resultout is again summed over all the documents.
            Calculate call the output as quantization error.
   Endloop
Endloop
```

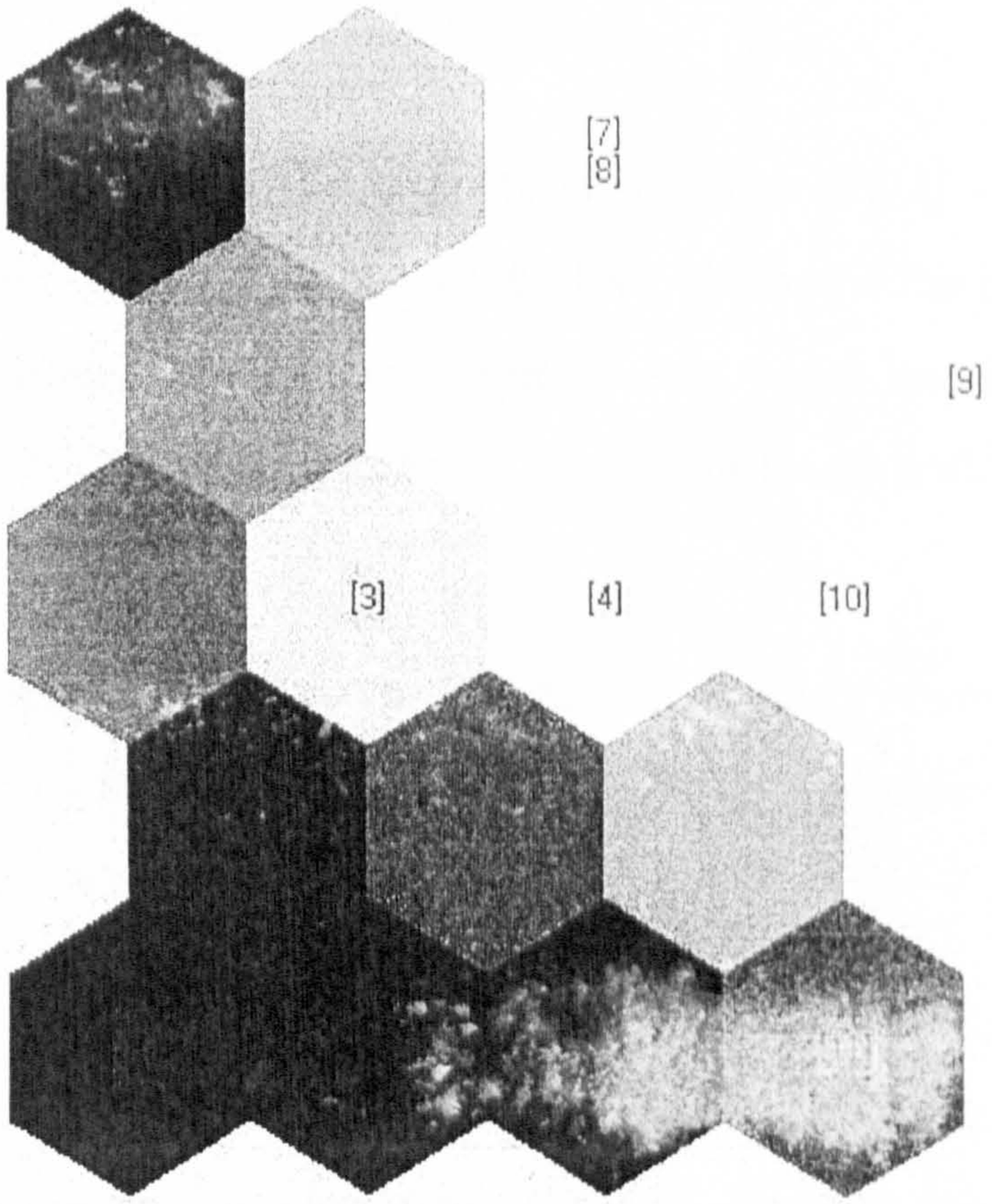


Figure 5.3: The Output of Training 12 documents

Figure 5.3 illustrates the $5X4$ visualization result grid of training the small corpus for Arabic and English documents onto a two-dimensional output space. They are translations of each other.

5.5 Stage II (Training SOM and Clustering)

In this stage we use the output of the first stage as the input in order to train the SOM network. We used a SOM 6X4 two-dimensional array. The number of input data was 20 documents in Arabic and 20 documents in English. In the ordering iteration, the number of the learning steps was set at 1000, the initial value of the learning rate $\alpha(0)= 0.5$ and the radius of the niegbourhood $R(0)=7$. The initial weight vectors w_{ij} were set randomly between 0 and 1.0. The procedure for form-

ing the document category map involved five basic steps, which are presented and explained below.

1. The model is implemented using Matlab program ($SOM - Get - Docs$) function to create a matrix of $Nword$ X $Mdoc$. $Nword$ is the number of words in every document, $Mdoc$ number of documents, which makes all columns identical in length, filling all columns of insufficient length with "10000000"". The words are encoded numerically.

2. Create a unique word matrix $Nword$ X 1, where $Nword$ is the number of unique words from all documents and contains one column. Therefore, the word does not repeat.

3. Create a matrix $Nword$ X $Ndoc$, where $Nword$ is the number of roots in every document, and $Ndoc$ the number of documents. The random codes formed in step 1 are used in the calculation. This step computes the frequency of every root in each document. Then, find the maximum occurrence of each word in each document. After that, find the index of these words. Finally, compare these words with the bilingual dictionary.

4. Add lables using function $SOM - Get - Catg$ for each word in the matrix WC_doc, and add a category for each vector by determining the frequency of the keyword in every document. Firstly, search the word number which has the highest frequency in the dictionary and get its category number. Secondly, determine the category number from the dictionary, then check if language index equals 1; if it is we have the Arabic category, otherwise English. See Figure 5.4. Finally, the system produces the final matrix $ASMA$ that consists of all vectors for Arabic-English documents. The $ASMA$ data set also has labels associated with the data. Actually, the data set consists documents in Arabic and English samples of 8 main categories of $ASMA$.

5. The matrix formed in step 4 is the input to the SOM. The resulting map is labeled after the training process by inputing the input vectors once again and by naming the best matching neurons according to the key word part of the input vector.

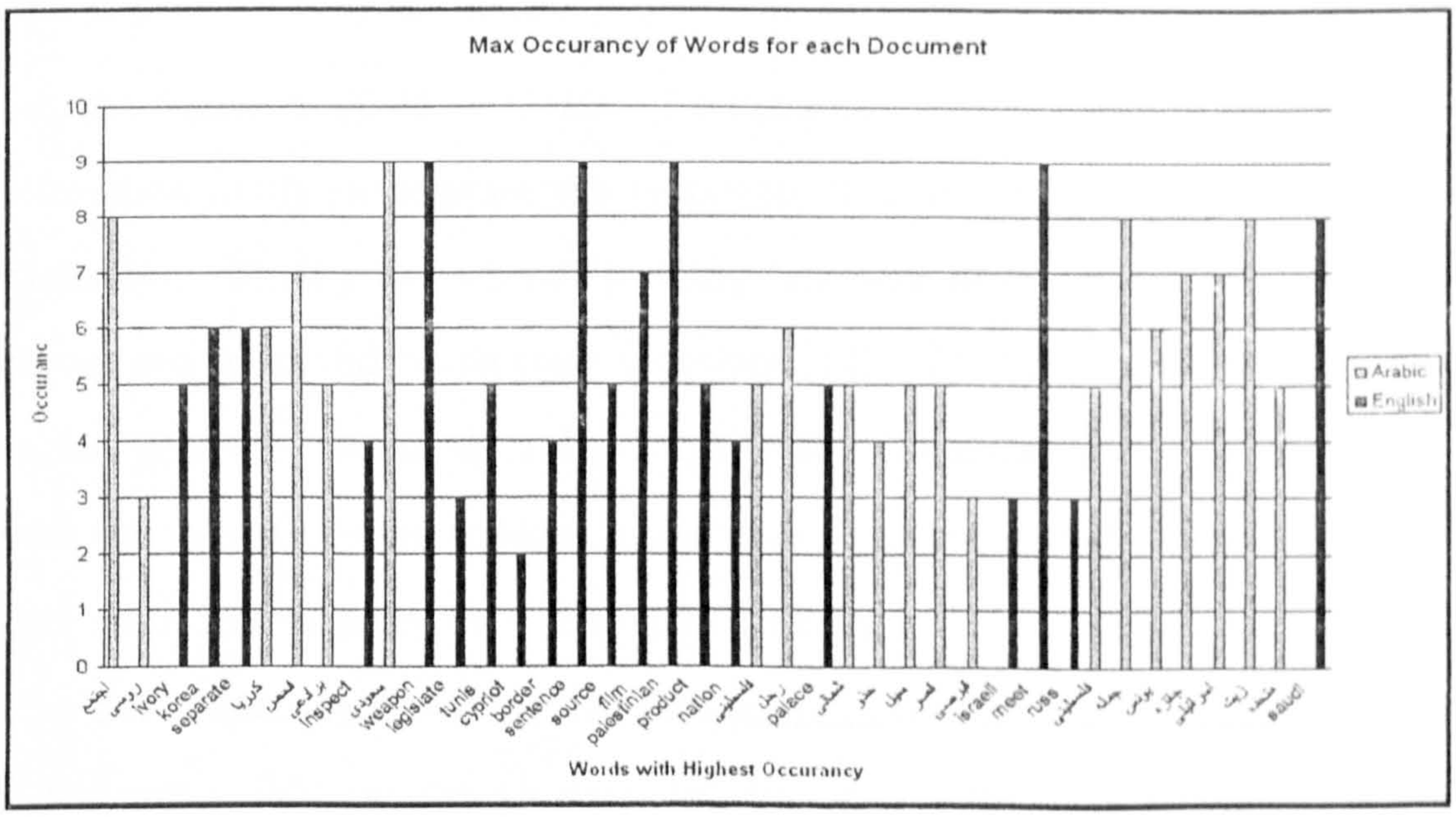


Figure 5.4: The Word with Max Frequency for Each Document

Figure 5.4 illustrates the important patterns in each document, highlighting the maximum occurrences from within the Arabic and English documents.

The basic idea in the SOM learning process is that for each sample input vector x_i^k. The SOM has a special structure, called data structure, which is used to group information regarding the data set in one place. Here, a data-struct is created using function $SOM - Data - Struct$. Firstly, the data matrix ASMA structured, then a name is given to the data set, and to the variables in the data matrix. Then the model reads the data directly from an ASCII file. Finally, the function $SOM - Normalize - Data$ is used to convert the data which is scaled to $(0 - 1)$ based on the highest word frequency in all documents. The data normalisation is defined as follow:

$$normalized(k,j) = \frac{WC_{doc}(k,j)}{maxwcdoc(k,j)}$$

where WC_{doc} is a matrix containing the frequency (number of root occurrences in each document), *maxwcdoc* is the maximum frequency in the dataset, and this makes it possible to train the SOM network.

Next, the function $SOM - Make - Struct$is used to train the SOM. In this step the function firstly determines the map size, then initializes the map using linear initialization. Finally two variant training functions have been implemented in the toolbox: seq-train and batch-train functions [149]. In the sequential training function, the data are presented to the map one at a time, and the algorithm gradually moves the weight vectors towards input data. The sequential training function requires much less memory than the batch training function for training the dataset. In the batch training function, the data set are presented to the SOM as a whole, and the new weight vectors are weighted averages of the data vectors. Both training functions are iterative, but the batch version is much faster in Matlab since matrix operations can be utilized efficiently. Since the SOM algorithm is based on Euclidean distances, calculating through Equation 3.3, the scale of the variables is very important in determining what the map will be like.

In the above, the initialization and the training were both done with the sequential training algorithm $SOM - Seqtrain$. In addition, the training was done in two phases: firstly with large neighborhood radius in the training, the map becomes stiffer and preserves the topology of the data set better.

5.6 Setup for Training Parameters

It is obvious that optimal parameters are different in each case, therefore, a number of recommendations for parameters have been applied in this training process [2].

These recommendations are starting points from which to work out the optimal parameters for the experiment in particular. When training small maps (less than a few hundred nodes), the selection of parameters does not greatly influence the outcome of the training process There are, however, a number of recommendations for training SOM which should be noted. These recommendations are discussed below:

The network topology refers to the shape of the grid, i.e. rectangular or hexagonal. The topology should in this case be hexagonal, since hexagonal grid is better for visualization purposes Network size, or the dimensions of the map, is important for visualization purposes. If the map is too small, differences between units are hard to identify. However, a small map is best for cluster identification purposes. On the other hand, if the map is too large, the clusters do not appear, and the map seems "flat".

The statistical accuracy of the mapping depends upon the number of steps in the final learning phase. This phase therefore has to be relatively large. The most important rule that should be applied in order to get the targeted result of iteration in the final phase, is that it must be at least 500 times the amount of nodes in the network It is common practice for the initial training phase to have at least 10% of the number of steps used in the final iteration.

The function $SOM - Train - Struct$ using to set the learning rate factor $\alpha(t)$, should start out as fairly large in the first iteration, but should be close to zero in final iteration. A commonly used initial learning rate is 0.5 for the first phase, and 0.05 in the fine-tune phase. The selection of the network neighbourhood size, $R(t)$, is possibly the most important parameter. If the selected neighbourhood size is too small, the network will not be ordered globally. Therefore, the initial network radius should be rather large, preferably larger than half the network diameter.

Like most NNs, the SOM has two ways of operation:

1. Within the training process, a map is built, and the NN organizes itself using a competitive process. The NN must be given a huge number of input vectors.

2. Within the mapping process, a new input vector may quickly by given a location on the map, it is automatically classified or categorized. There will be one single winner neuron; the neuron whose weight vector lies closest to the input vector by using Euclidean distance between input vectors and weight vectors.

Next we compared the performance of the model with different initial values of learning rate. In literature various initial guesses from 0.4-0.8 for the learning rate are used [86, 150, 17]. In our experiment a single data set of Arabic-English corpus composed of 20 documents was used as a test base and the initial learning rate is varied from 0.9 to 0.2. Results obtained are displayed in the tabular form as well as in graphical form (see Table 5.2, Figure 5.6 and Figure 5.5).

The performance of the model with evolving initial learning rate is measured via the quantization error and the CPU-time (measured in seconds). Results revealed a set pattern of monotonically increasing quantization error and a decreasing profile of the CPU time with drop in the learning rate in steps.

It is thus apparent that an optimized value for the learning rate must be selected which could bargain for relatively small CPU-time with acceptable quantization error. It is interesting to notice that at learning rate of 0.5 a sudden drop in the CPU-time is observed. Hence a value in between 0.6-0.5 would be a plausible choice.

Table 5.2: Results of Different Learning Rate values

Initial L-R	Documents	Radius	Clusters	Quantization Error	CPU-Time
0.9	20	6	3	6.38	71.16
0.8	20	6	2	6.40	71.12
0.7	20	6	4	6.42	71.10
0.6	20	6	4	6.45	71.08
0.5	20	6	4	6.46	42.68
0.4	20	6	3	6.48	41.50
0.3	20	6	3	6.52	40.45
0.2	20	6	3	6.61	40.39

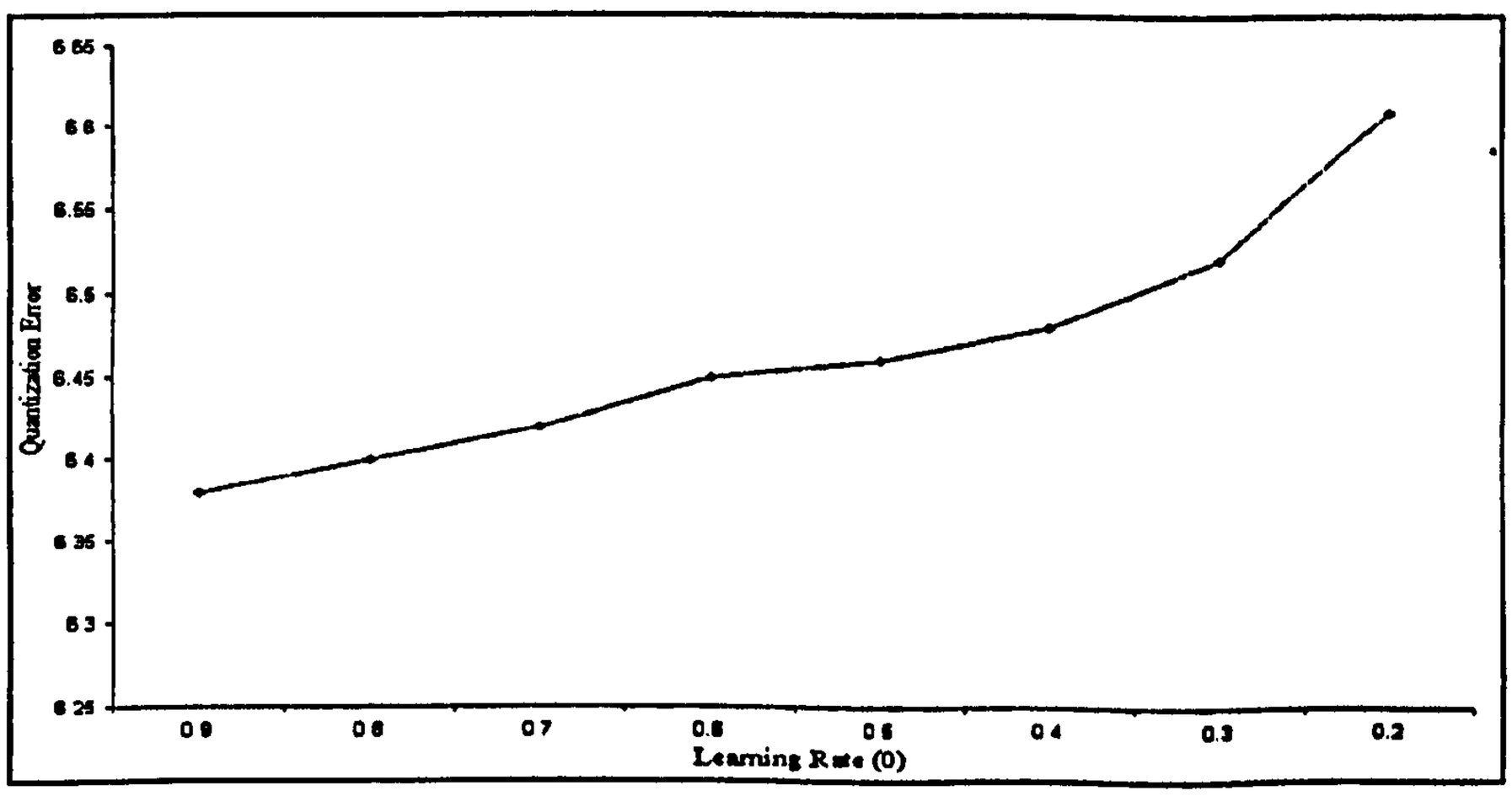


Figure 5.5: Execution Times of Different Learning Rate

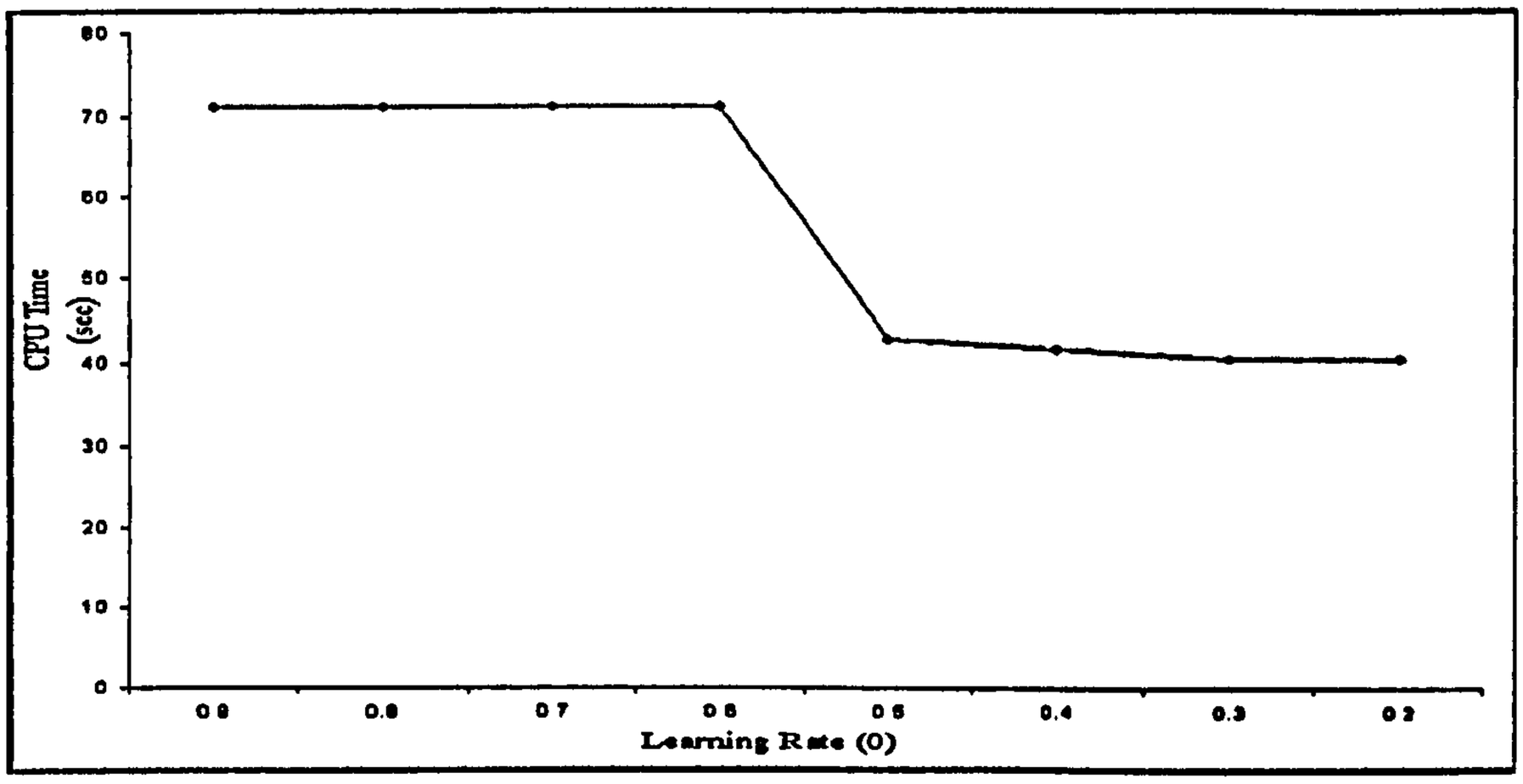


Figure 5.6: Performance of the model is tested with decreasing learning rate. Vertical axis shows the CPU-time consumed at each initial value of the learning rate.

5.7 The Program Packages Used

In this thesis we have introduced the SOM Toolbox for Matlab 5 [5], as the software package used for training the SOM, or SOM_PAK. SOM toolbox is a program package developed by the Neural Networks Research Center (NNRC) at the Helsinki University of Technology. The program package is free for non-commercial use. The version of SOM Toolbox used in this experiment is version 7.01.(R14). SOM Toolbox consists of a number of separate programs used for the training process. Each step of the process can be run using separate programs. Therefore, all maps displayed in this thesis will be displayed using SOM Toolbox.

The main function $SOM - DOCS - ASMA$ of this package has been utilized with essential modification to adapt to the Arabic language, see the flowchart of the programs are used in this thesis in Figure 5.7. Function $SOM - Get - Catg$ contains all categories named in Arabic-English languages. Then, $SOM - Get - Dict$ function download the dictionary in the Matlab environment. Finally, the function $SOM - Get - Docs$ prepared the data from output file "writedoc.txt" to next stage (see the Matlab code in Appendix C). The original output of SOM Toolbox was

displayed only English words but our modified output provides in Arabic script. Also, we modified the structure of the code in order to read the whole documents and adding the category belongs to each document for both languages from the categories database. Finally, the output of this tool was clustered and illustrated in the SOM grid.

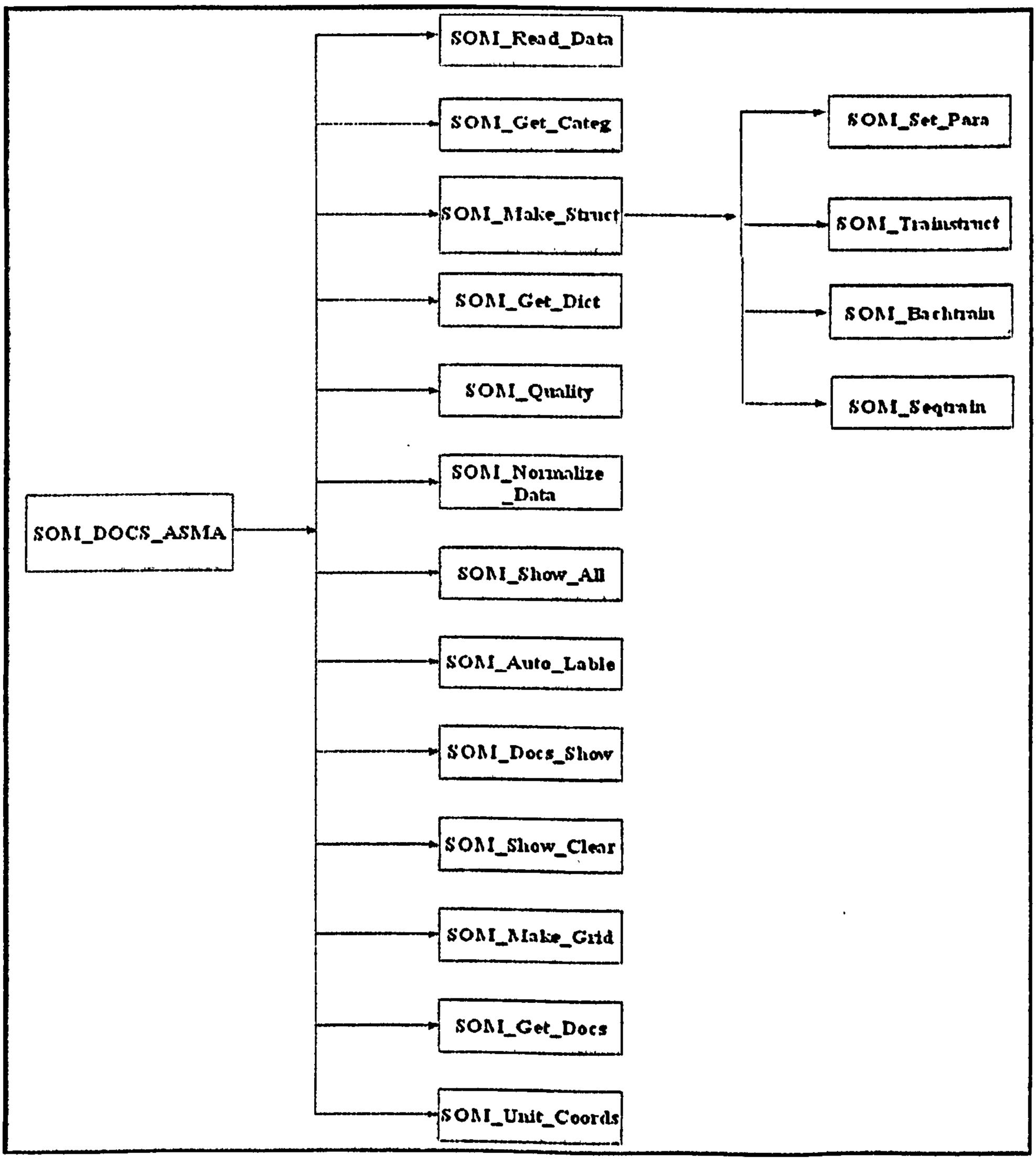


Figure 5.7: Flowchart of Matlab Programs

5.8 Stage III (Quality of Test and Data Visualization

In this thesis, we evaluate to what extent users of the SOM technique are satisfied with this technique in visualizing large amounts of data. The SOM method is a special type of neural network used in clustering and visualization.

5.8.1 Quality of Test

SOM has two main qualities [149]:

1. Classification accuracy.

   Bias and precision are both involved in accuracy when building a statistical model, but these need to be clear as accuracy may often be only achieved by carefully balancing the two. Accuracy can be used to describe the degree of "correctness" of a map or classification, especially in thematic and classified mapping using remotely accessed data. Such a map may be considered accurate if it offers an unbiased representation of the region it is supposed to portray. Thus, classification accuracy is a measure of how closely a modelled map conforms with real-world actuality [151].

2. Data visualization accuracy.

   Data visualization is considered to be a powerful tool for examining and interpreting raw data. Quantitative information is usually best understood as images, and data visualization helps to more easily reveal information that is inherent but hidden within large bodies of data. A researcher can map data features with geometric attributes by exploiting known details such as length, area or positions. Histograms and bar charts are often effectively employed, showing quantity as length, as are scatter plots, showing how variables are

connected by highlighting density and location [152].

Furthermore, the quality can be tested to determine whether more training is needed or if the clusters are acceptable. When text mining large amounts of information, the scarcity of suitable tools for analysis is apparent. However, the SOM, which has been applied to a wide range of problems, specifically to multilingual language problems [17], has the quality that any given map may be measured for statistical accuracy through Quantization Error (QE), given by equation (3.5).

5.8.2 Data Visualization

Once a SOM has been created, it must be visualized in order for it to be interpreted. Unified distance matrix [102] is the most common way for visualizing SOM. In order to create the maps, one must calculate the average of the distances of a reference vector in a node to that of its neighbouring reference vectors. This average is placed at the appropriate coordinate on the matrix. Needless to say, the shape of the matrix is dependent upon the neighbourhood topology, i.e. rectangular or hexagonal. Therefore, Figure 5.8 illustrates the clusters with lightly shaded areas representing the short distances between them, while the clusters with dark shaded representing long distances. This effectively allows us to locate similar units on the map, and to identify groups of similar units. This process can be called clustering via visualization [153], i.e. the SOM is used to cluster the data, and the clusters are subjectively isolated by studying the visualization of the topology. The SOM network allows the visualization function in freely specified coordinates, for example the input space. Basically, the $SOM - Grid$ function visualizes the SOM network, where each node is connected to its neighbour with weight.

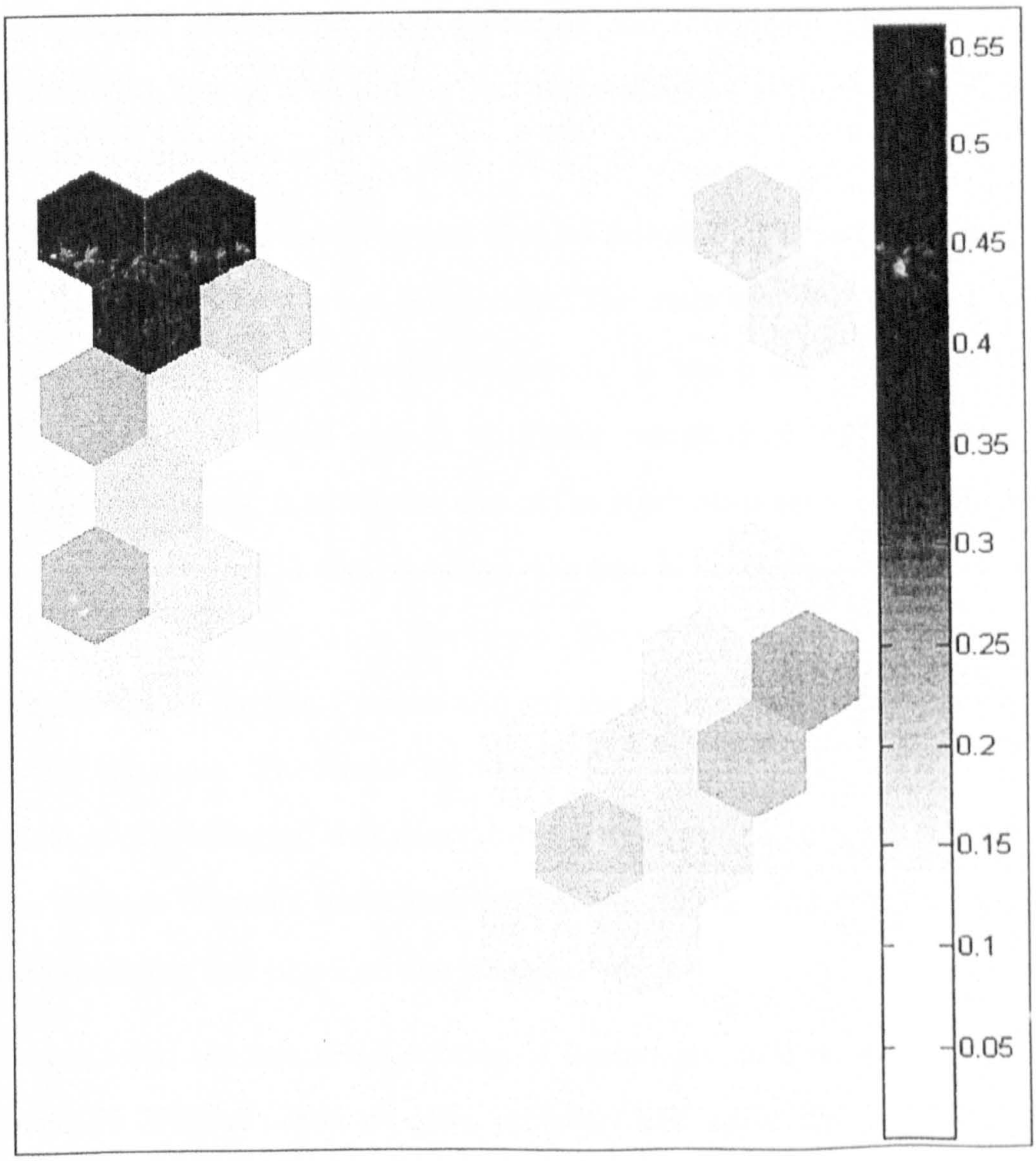


Figure 5.8: Visualization the Clusters on the Map

5.9 Summary

This chapter has identified and presented the approaches for multilingual text mining (MLTextMAES). The required pre-processing stages and methods for Arabic-English, and the Arabic-English stemming system were all presented. General-purpose bilingual dictionaries were developed and introduced. Finally, the algorithms for multi-lingual text mining for Arabic-English (MLTextMAES) and the SOM network were developed.

The morphological analyser uses rules of word formation (derivation morphology). This processor is at the heart of the system. The main aim of developing this processor is to support the root method of search. It was pointed out here that this processor does not cover all aspects of Arabic morphology, such as phonological changing or verb forms; it is not the aim of the study to cover all aspects of Arabic morphology. In the lexical analysis stage, the text is broken down into tokens, and then a general stop word list is identified. Stop words are then deleted thereby reducing the size of the file. Prefixes and suffixes are removed to find the stems and then to find the roots. The Arabic root-based algorithm was detailed. The semantic unification of the bilingual dictionary involved assigning indices to roots that are common through Sharoff's text classification. We also demonstrated an example of this pre-processing and provided the indices.

We demonstrated the automatic mining of documents, and showed that the algorithm consists of three parts: analysis, structure and visualization. The final stage is presented as a 6X4 grid. When a fully-trained algorithm is performing well in an SOM, the output feature map will have clustered the data into a two-dimensional output space, in such a way that all similar nodes are located within their own cluster, and all relatively similar clusters are located relatively closely. Furthermore, all distances, within and between clusters will be a reflection of similarity. This was successfully demonstrated here where the training was unsupervised and the learn-

ing rate was reduced decreases geometrically. The input was effectively reduced in dimensionality and the output of our SOM is clear.

One of our contributions to this work consists of identifying the factors that affect quality when using the SOM technique in the visualization of data, and of how to utilize the information obtained from training. We were able to show that the SOM is good at statistical measurements and provides accurate data visualization from the results of clustering. This chapter represents the conceptual framework of the study; the implementation of the framework is discussed in the next chapter.

Chapter 6

Implementation and Experiments

Objectives

- To present the SOM network design.
- To implement SOMMLTM algorithm.
- To present the software package used.
- To demonstrate the clusters.

6.1 Introduction

Chapter 5 has already presented details on our prototype for a multilingual text mining model in connection with knowledge mining techniques. In this chapter we discuss a variety of details concerning the implementation of our MLTextMAES model. The codes were implemented in Java and Matlab programming languages; the model compiles and executes within these environments. The implementation of the MLTextMAES consists of three parts. First, pre-processing, where filtering

and explanation engines enable applications to filter a set of documents using Porter Stemmer and AraMorph stemmer and give explanations about filtering decisions. Second, generating numeric matrix to train the retrieved data. Finally, training the self-organizing multilingual map technique for Arabic-English corpus and constructing maps of documents. Moreover,the algorithm is also assessed by measuring the performance of the trained data. The algorithm for multilingual text mining using SOM is conducted in three steps. In the first step, the corpus for Arabic-English is analysed in order to generate the indices for every root, as explained in Section 5.1.8. The second step is to construct the structure dataset, ASMA. Finally, it is essential to visualize the output in the grid by training the SOMMLTM algorithm.

6.2 Stage I (Pre-Processing)

This is the main stage of the framework; it composed of the following components, (1) stop words removal, (2) prefixes and suffixes removal, (3) infixes removal, (4) semantic unification of bilingual dictionary, (5) indices generation. Each component is implemented in the manner discussed below; see also Figure 6.3.

Pre-processing has been applied using the Arabic algorithm for a Root-Based stemming approach (as in Section 5.2) together with English Porter Stemmer (Stemmer) [113, 138]. Significant modifications in the Arabic Morphology and English are needed in order to put the stemmer program into action. The outputs of the original Arabic Morphology were Arabic words in English alphabets with a long chain of possible English meanings. But our modified outputs offer only the most plausible. Moreover, the English stemmer was capable of reading only a single English word and stemming that. We modified the structure of the code in order to read the whole document, eliminating the stop words from the document and generating the keywords of the document. For examples of the original corpus for Arabic-English, see Figure 6.1, and Figure 6.2 below.

Figure 6.1: (a) The Original Arabic Corpus

File Edit View Insert Format Help

>DOC docid="AFA20030101.5900"
sysid="ahd><"hl><seg id=1> Izzet Ibrahim Meets Saudi Trade Official in Baghdad </seg/><hl><p><seg id=2
/>p<p<seg id=3> The agency said Ibrahim welcomed this occasion for trade exchange and cooperation betw
Iraq and Saudi Arabia. </seg/><p<p<seg id=4> The agency also reported that the Iraqi Minister of Trade
Mohamed Mehdi Salih, took part in the meeting. </seg/><p><<seg id=5> Baghdad and Riyadh, who broke di
relations during the Gulf War in 1991, began to improve their relations over the course of the Beirut
last March. </seg/><p<p><seg id=6> Saudi sources this week denied reports in the American New York Tim
that Saudi Arabia had agreed to allow the United States to use Saudi military bases should a war agai
Iraq take place. </seg/><p><p><seg id=7> Prince Abdulrahman bin Azziz, Saudi Arabia's Defense Minister
said in a statement to Saudi newspaper Ugath that 'what was claimed by the newspaper is incorrect.
' </seg/><p><p><seg id=8> He added: 'The Kingdom's position has been clear since the beginning and we
not able to place our airspace and bases' under American control. </seg/><p><p><seg id=9> Saudi Arabia
a base for American forces during the Gulf War (January and February 1991). </seg/><p/<DOC<
>DOC docid="AFA20030102.4200"
sysid="ahd><"hl><seg id=1> Israeli Army Destroys 10 Palestinian Homes during Incursion into Rafah
</seg/><hl><p><seg id=2> Gazza 1-2 (AFP) - An official security source stated today that the Israeli
army destroyed at least 10 Palestinian homes on Wednesday night during an incursion into Rafah Palesti
Refugee Camp in the southern Gazza Strip. </seg/><p><p><seg id=3> The source told France Presse:
'The occupation forces backed by 12 tanks, military mechanized equipment and 2 bulldozers penetrated o
200 meters into Palestinian land early this morning in the Block G area of Rafah Refugee Camp, comple
destroying 10 houses and damaging 10 other houses in the midst of heavy gunfire.' </seg/><p><p><seg id
He said that 'Israeli soldiers opened heavy fire on houses in the Camp, forcing their inhabitants to l
' </seg/><p><p><seg id=5> Witnesses said armed skirmishes took place between armed Palestinians and th
Israeli army in the Camp, although no information concerning casualties is available.
</seg/><p/><DOC<
>DOC docid="AFA20030103.6000"
sysid="ahd><"hl><seg id=1> Iraqi Opposition Prepares for mid-January Meeting in Kurdistan </seg<
/>hl><p><seg id=2> London 1-4 (AFP) - Seven Iraqi opposition groups held a meeting in London last nigh
/>p><p><seg id=3> Ahmed al-Bayati, a member of the Committee, told France Presse that last night's mee
g in London was connected with the conference organized by 7 Iraqi opposition movements in London last
December 14-17 in the course of which it was agreed that this meeting should take place in Iraqi Kurdi
</seg/><p><p><seg id=4> He added that the last night's agenda was to 'discuss the Committee's meeting
to be held mid-January in Iraqi Kurdistan' defining organizational and logistical preparations.
</seg/><p><p><seg id=5> He said the committee would form sub-committees during the meeting in Kurdist
and was preparing the opposition's schedule, adding that the agenda had not yet been decided.
</seg/><p><p><seg id=6> He said the final decision concerning the conference's place and date would be
over the course of another meeting to be held early next week. </seg/><p/><DOC<

Figure 6.2: (b) The Original English Corpus

The Figure 6.1 above presents a portion of the original corpus before pre-processing, containing HTML tags, articles, conjunctions, prepositions, punctuation and auxiliary verbs etc.

We selected 20 Arabic documents as the test case. The MMA was then executed to provide the final result as a text file and convert then the roots back into codes. These documents are related to five main categories (Commerce, Life, Politics, Social Sciences, and Entertainment).

The Arabic portion in Figure 6.2 show that the original 20 documents are the exact translation in English before any processing is done. In the pre-processing stage, the available texts from both languages are read and are then lexically analysed and resolved into words. Throughout the pre-processing stage, the filtering process is generated and applied to the text in order to eliminate stop words such as articles, conjunctions, preposition, punctuation and auxiliary verbs etc. (examples are: he, is, and, in, the, !, .etc).

The morphological analyser uses rules of word formation, in other words, derivation morphology. This process is at the heart of the system and consists of several methods. The model is illustrated in Figure 6.3.

The major goal of developing this process is to support and improve the root method of search, not to cover all aspects of Arabic morphology, such as phonological changing and verb forms. The root method is a novel approach and has been introduced into this study as we believe it to be a major element in improving the mining of text in Arabic. The following sections discuss the steps involved in finding the roots.

6.2.1 Lexical Analysis

In this component the texts in both languages are read, and then lexically analysed and resolved into words. The system converts the letter before stress into a

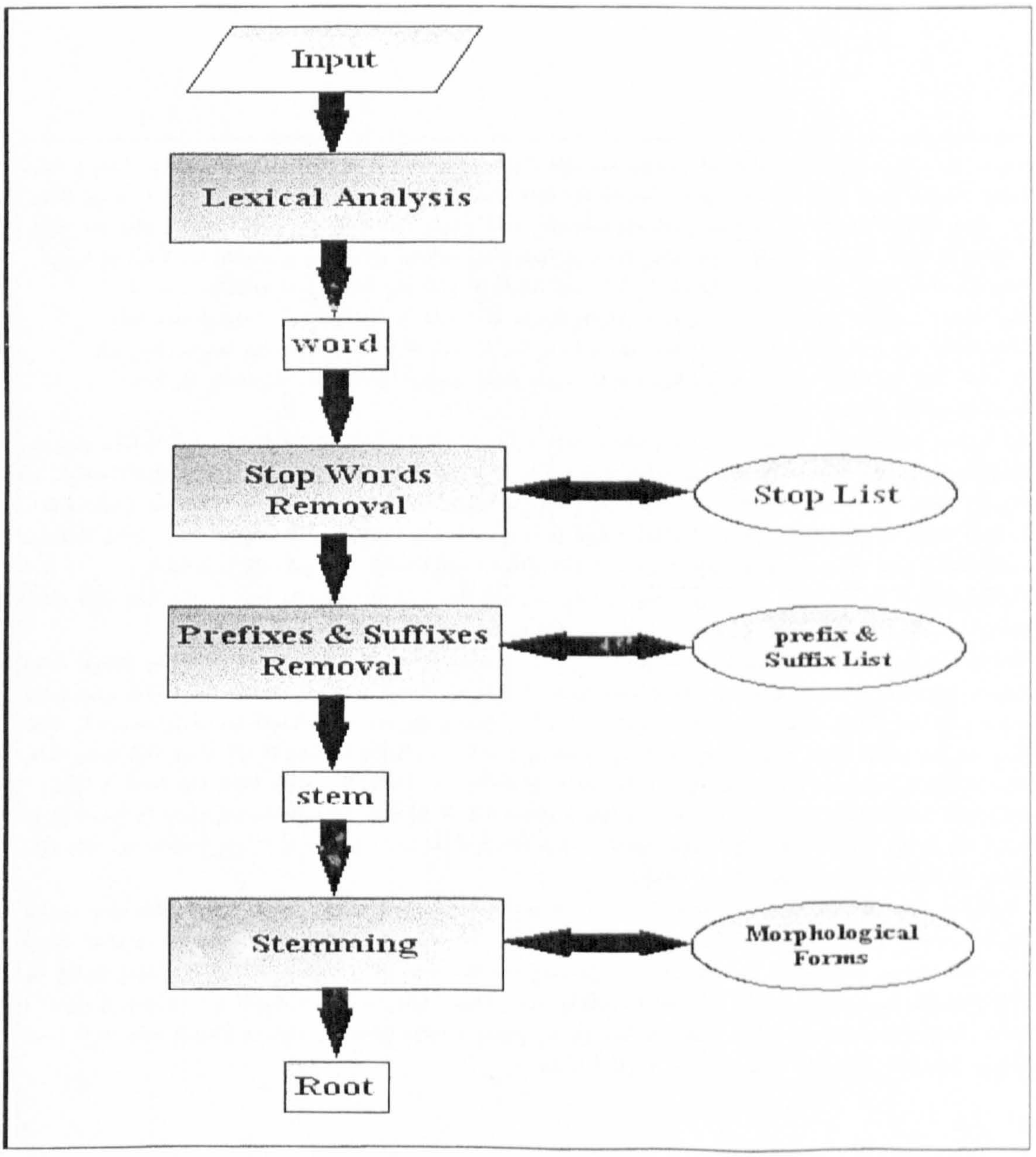


Figure 6.3: Morphological Analysor

duplicate, changes the expanded alif آ to alif ا, and deletes all HTML tags and punctuation, as in the example shown below. Those words are then passed through the filtering process i.e. eliminating stop words, prefixes and suffices. The results are shown in Figures 6.4, and (6.5 below.

المتمردو قرب ساحل العاج قرروا مقاطعة مؤتمر باريس حول ازمة البلاد ابيدجان اف فررت الحركة الشعبية لساحل العاج في الغرب الكبير المتمردة اليوم الجمعة مقاطعة الاجتماع السياسي حول ازمة ساحل العاج الذي سيفتتح اعماله في باريس في ١٥ كانون الثاني الر مجمات القوات الحكومية على مواقعها في تولبيلو وبلوليكان وذلك في بيان وصلت نسخة منه الىوكال الناطق باسم هذه الحركة م فياتو ان الحركة تعتبر ان اجتماع باريس انتهى اليوم مع الهجمات على مواقعها في تولبيلو وبلوليكان على الحدود مع ليبيريا الغربي قرب ساحل العاج واضاف ان هذه الهجمات التي تاتي بعد اقل من ٤٨ ساعة على اعلان حركتنا موافقتها المبدئية على التوجه الى باريس للمشاركة في ايجاد حل سياسي للازمة، تخطرنا للرد وبعد ان اكد ان القوات الحكومية قصفت صباح الخميس منطقة فرابو على بعد حوالى ٢٠٠ كلم جنوب بلوليكان ما ادى الى مقتل ١٥ مدنيا واصابة ثلاثة متمردين بجروح قال ان الحركة تعتبر انها تعرضت للخيانة من قبل فرنسا ودعا البيان الحركة الوطنية لساحل العاج التي تسيطر على شمال البلاد الى مقاطعة الاجتماع ايضا

غزة ابراهيم بمستقبل مسؤولا اقتصاديا سعوديا في بغداد بغداد ١ اف ب ذكرت وكالة الانباء العراقية الرسمية ان نائب رئيس مجلس الجيش الاسرائيلي يهدم عشرة منازل فلسطينية خلال توغل في رفح غزة ٢ ١ افاد مصدر امني رسمي اليوم الخميس ان الجيش الاسرائيلي هدم الاربعاء الخميس عشرة منازل على الاقل خلال عملية توغل في مخيم رفح للاجئين الفلسطينيين جنوب قطاع غزة وقال المصدر لوكالة فرانس ان قوات الاحتلال هدمت بالتي عشرة دبابة وآلية عسكرية وجرافتين توغلت فجر اليوم اكثر من مائتي متر في عمق الاراضي الفلسطينية منطقة بلوك ج بمخيم رفح ودمرت عشرة منازل تدميرا كاملا والحقت اضرارا بعشرة منازل اخرى وسط اطلاق كثيف للنار واشار الى ان ا اسرائيليين فتحوا النار بكثافة على منازل المخيم ما اضطر سكان هذه المنازل الى مغادرتها واكد شهود ان اشتباكات مسلحة وقعت ب مسلحين فلسطينيين والجيش الاسرائيلي في المخيم لم تتوفر معلومات عن وقوع اصابات

المعارضة العراقية تعقد اجتماع يعقد منتصف كانون الثاني يناير في كردستان لندن ٤ ١ اف ب دت صبح مجموعات من المعارضة العراق اجتماعا امس الجمعة في لندن للاعداد لاول اجتماع للجنة المصغرة المؤلفة من ٢٥ عضوا والتي تشكلت منتصف كانون الاول ديسمبر في لندن سيعقد في كردستان العراق منتصف كانون الثاني يناير وقال احمد البياتي وهو احد اعضاء اللجنة لوكالة فرانس برس ان الاجتما الذي عقد امس في لندن متعلق بالمؤتمر الذي نظمته صبح حركات من المعارضة العراقية في لندن بين ١٤ و١٧ كانون الاول ديسمبر الماضي خلاله الاتفاق على مبدأ عقد هذا الاجتماع في كردستان العراق واضاف ان جدول اعمال اجتماع امس الجمعة كان البحث في اجتماع ا الذي سيعقد منتصف كانون الثاني يناير في كردستان العراق وتحديدا الناحية التنظيم والاعداد اللوجستي واوضح ان اللجنة قد تنش فرعية خلال اجتماع كردستان وتعد عمل المعارضة موضحا ان جدول الاعمال لم يقر بعد واشار الى ان القرار النهائي حول مكان وزما سيتخذ خلال اجتماع اخر يعقد مطلع الاسبوع المقبل

مقتل ثمانية بينهم ستة من الروم في حادث سير الناصرة ٥ ١ أف ب ذكرت وكالة انباء الشرق الاوسط ان ثمانية اشخاص قتلوا، بينهم ستة من الروم واصيب ١٢ اخرون بجروح في حادث سير وقع في جنوب سيناء وفقا لاخر حصيلة اليوم الاحد واضافت الوكالة ان حافلة سياحية صباحا اصطدمت امام الصبت بصبارة خاصة وصبارة اجرة قرب منتجع دهب الذي يبعد مسافة ١٠ كلم من صنجع شرم الشيخ الواقع على الاحمر والضحايا الروم هم رجل وزوجته وابنتهما البالغة من العمر ثمانية اعوام اضافة الى ثمانين ١١ و ٧ اعوام ورجل وزوجته و الحافلة المصري الجنسية اما الجرحى فهم اربعة بريطانيين وفرنسي وسويدي وروماني وبلجيكيان وهولندي ومصريان بينهم المرشد السياح واظهرت التحقيقات الاولية ان الحادث ناجم عن السرعة الفائقة

Figure 6.4: (a) Arabic Corpus After a Lexical Analysis

These Figures 6.4, and 6.5 demonstrate the Arabic-English corpus without the HTML tags and punctuation, etc., this will be the input to next component in

File Edit View Insert Format Help

Izzet Ibrahim Meets Saudi Trade Official in Baghdad Baghdad 11 AFP Iraq official news agency reported that the Deputy Chairman of the Iraqi Revolutionary Command Council Izzet Ibrahim today met with Abdul Rahman al-Zamil Managing Director of the Saudi Center for Export Development The agency said Ibrahim welcomed this occasion for trade exchange and cooperation between Iraq and Saudi Arabia The agency also reported that the Iraqi Minister of Trade Mohamed Mehdi Salih took part in the meeting Baghdad and Riyadh who broke diplomatic relations during the Gulf War in 1991 began to improve their relations over the course of the Beirut Summit last March Saudi sources this week denied reports in the American New York Times that Saudi Arabia had agreed to allow the United States to use Saudi military bases should a war against Iraq take place Prince Abdulrahman bin Azziz Saudi Arabia's Defense Minister said in a statement to Saudi newspaper Uqath that what was claimed by the newspaper is incorrect He added The Kingdom's position has been clear since the beginning and we are not able to place our airspace and bases under American control Saudi Arabia was a base for American forces during the Gulf War January and February 1991
Israeli Army Destroys 10 Palestinian Homes during Incursion into Rafah Gazza 12 AFP An official security source stated today that the Israeli army destroyed at least 10 Palestinian homes on Wednesday night during an incursion into Rafah Palestinian Refugee Camp in the southern Gazza Strip The source told France Presse The occupation forces backed by 12 tanks military mechanized equipment and 2 bulldozers penetrated over 200 meters into Palestinian land early this morning in the Block G area of Rafah Refugee Camp completely destroying 10 houses and damaging 10 other houses in the midst of heavy gunfire He said that Israeli soldiers opened heavy fire on houses in the Camp forcing their inhabitants to leave Witnesses said armed
Eight Killed including 7 Russians in Traffic Accident Cairo 15 AFP Al-Sharq Al-Awsat news agency reported that 8 people including 7 Russians were killed and 12 others injured in a traffic accident in southern Sinai according to the last count It added that a tourist coach carrying tourists collided with a private car and a rented car near the Dahab Resort which lies about 60 kilometers from the resort of Sharm el-Sheikh on the Red Sea The Russian victims were a husband and wife and their 8 year old daughter two young girls 11 and 7 years old a man and his wife and the Egyptia coach driver 4 British 1 French 1 Swedish 2 Belgian 1 Dutch and 2 Egyptian tourists including the guide were injured I West Ivory Coast Rebels Decide to Boycott Paris Summit on Country's Crisis Abidjan 110 AFP According to a statement received in Abidjan, the rebel group the Popular Movement for the West Ivory Coast decided today to boycott the politi January 15 after government forces attacked their positions in Toliblu and Blolikan The movement's spokesman Ghioum Gabato said The movement regards the Paris conference as having ended today with the attacks on its positions in Tolil and Blolikan on the border with Liberia furthest point west on the Ivory Coast He added These attacks coming less thai 24 hours after our movement announced its agreement in principle to go to Paris to take part in finding a political solution to the crisis, obliges us to respond After confirming that government forces had bombed the Grabu region some 200 kilometers south of Blolikan killing 15 civilians and injuring 3 rebels he said the movement considers it has been betrayed by France The statement called on the Ivory Coast Nationalist Movement which controls the north of the country to boycott the conference too

Figure 6.5: English Corpus After a Lexical Analysis

this stage.

6.2.2 Prefixes and Suffixes Removal

This process is to remove the prefixes and suffixes attached to words. To do this, two lists of prefixes and suffixes are created. Then the attached prefixes and suffixes are removed from each word. The lists of prefixes and suffixes are illustrated in Figure 6.6, and Figure 6.7 below, and the result of this process is stems only, as in Table 6.1. This result will be the input to the next process.

Table 6.1: An Example of Stemming in Arabic Language

	Word	Prefixes	Stem	Suffixes
Example	وليجمعكم	ول	يجمع	كم
Transliteration	walyjmEkum	wl	yjmE	kum
Meaning	to combines you	to	Combines	you

6.2.3 Stemming In Arabic

This component has been implemented using the Arabic stemming algorithm AMESD, which is written in Java language. The AMESD will eliminate the prefixes, suffixes and infixes from each word to obtain the roots as shown in the example of the word "surrender,تنازل", first remove the prefix "ت", then remove the infix "ا", finally, the reminder will be the root "descend,نزل". See the result in Figure 6.8 below.

The Arabic stemmer was applied to the Arabic corpus and the results are saved in a single text file "read.txt" containing the roots from all the Arabic and English documents, see in Figure 6.8 below.

```
; conjunctions
w       wa      Pref-Wa   and <pos>wa/CONJ+</pos>
f       fa      Pref-Wa   and;so <pos>fa/CONJ+</pos>
;
; prepositions
; incompatible with noun suffix categories that are "nominative"
b       bi      NPref-Bi  by;with <pos>bi/PREP+</pos>
k       ka      NPref-Bi  like;such as <pos>ka/PREP+</pos>
; concatenations
wb      wabi    NPref-Bi  and + by/with <pos>wa/CONJ+bi/PREP+</pos>
fb      fabi    NPref-Bi  and + by/with <pos>fa/CONJ+bi/PREP+</pos>
wk      waka    NPref-Bi  and + like/such as <pos>wa/CONJ+ka/PREP+</pos>
fk      faka    NPref-Bi  and + like/such as <pos>fa/CONJ+ka/PREP+</pos>
;
; preposition li- (precedes nouns)
; incompatible with noun suffix categories that are "nominative"
l       li      NPref-Li  for/to <pos>li/PREP+</pos>
; conj. + prep. li-
wl      wali    NPref-Li  and + for/to <pos>wa/CONJ+li/PREP+</pos>
fl      fali    NPref-Li  and + for/to <pos>fa/CONJ+li/PREP+</pos>
;
; emphatic particle la- (precedes nouns)
; incompatible with suffix feature "genitive/accusative"
l       la      NPref-La  indeed/truly <pos>la/EMPHATIC_PARTICLE+</pos>
;
; result clause particle (laam jawaab al-shariT -- precedes perfect verb)
l       la      PVPref-La would have <pos>la/RESULT_CLAUSE_PARTICLE+</pos>
;
; definite article (incompatible with all poss.pron. sufixes)
Al      Al      NPref-Al  the <pos>Al/DET+</pos>
; conj. + def.art
wAl     waAl    NPref-Al  and + the     <pos>wa/CONJ+Al/DET+</pos>
fAl     faAl    NPref-Al  and/so + the  <pos>fa/CONJ+Al/DET+</pos>
;
; prep. + def.art.
```

Figure 6.6: A list of Prefixes in Arabic

File Edit Format View Help

```
;thun    atuhuun    NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+huu:POSS_PRON_3MP</pos>
;thA     atuhA      NSuff-ath its:their:her <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+hA:POSS_PRON_3FS</pos>
;thn     atuhun-a   NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+hun-a:POSS_PRON_3FP</pos>
;tk      atuka      NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+ka:POSS_PRON_2MS</pos>
;tk      atuki      NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+ki:POSS_PRON_2FS</pos>
;tkmA    atukumA    NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+kumA:POSS_PRON_2D</pos>
;tkm     atukum     NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+kum:POSS_PRON_2MP</pos>
;tkn     atukun-a   NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+kun-a:POSS_PRON_2FP</pos>
;nA      atunA      NSuff-ath our        <pos>+ap:NSUFF_FEM_SG+u:CASE_DEF_NOM+nA:POSS_PRON_1P</pos>
;;
;h       atahu      NSuff-ath his:its    <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+hu:POSS_PRON_3MS</pos>
;hmA     atahumA    NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+humA:POSS_PRON_3D</pos>
;hm      atahum     NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+hum:POSS_PRON_3MP</pos>
;hA      atahA      NSuff-ath its:their:her <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+hA:POSS_PRON_3FS</pos>
;hn      atahun-a   NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+hun-a:POSS_PRON_3FP</pos>
;k       ataka      NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+ka:POSS_PRON_2MS</pos>
;k       ataki      NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+ki:POSS_PRON_2FS</pos>
;kmA     atakumA    NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+kumA:POSS_PRON_2D</pos>
;km      atakum     NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+kum:POSS_PRON_2MP</pos>
;kn      atakun-a   NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+kun-a:POSS_PRON_2FP</pos>
;nA      atanA      NSuff-ath our        <pos>+ap:NSUFF_FEM_SG+a:CASE_DEF_ACC+nA:POSS_PRON_1P</pos>
;;
;h       atihi      NSuff-ath his:its    <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+hu:POSS_PRON_3MS</pos>
;hmA     atihimA    NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+humA:POSS_PRON_3D</pos>
;hm      atihim     NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+hum:POSS_PRON_3MP</pos>
;hA      atihA      NSuff-ath its:their:her <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+hA:POSS_PRON_3FS</pos>
;hn      atihin-a   NSuff-ath their      <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+hun-a:POSS_PRON_3FP</pos>
;k       atika      NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+ka:POSS_PRON_2MS</pos>
;k       atiki      NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+ki:POSS_PRON_2FS</pos>
;kmA     atikumA    NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+kumA:POSS_PRON_2D</pos>
;km      atikum     NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+kum:POSS_PRON_2MP</pos>
;kn      atikun-a   NSuff-ath your       <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+kun-a:POSS_PRON_2FP</pos>
;nA      atinA      NSuff-ath our        <pos>+ap:NSUFF_FEM_SG+i:CASE_DEF_GEN+nA:POSS_PRON_1P</pos>
```

Figure 6.7: A list of Suffixes in Arabic

Figure 6.8: Arabic Corpus After Stemming

6.2.4 Stemming In English

In the case of the English texts, the documents are read and passed through the analyser i.e. eliminating prefixes and suffixes. Moreover, after removing the prefixes and suffixes, an indexing procedure tries to conflate word variants into the same stem or root using a stemming algorithm. When applying the stemming algorithm, the approach is to try to build the basic forms of the words (root) by removing the suffix "s" from plural word forms, the "ed" which is the past participle ending, and the "ing" of the gerund ending from the verbs. The results may be viewed in Figure 6.9 below.

6.2.5 Arabic Morphology and English Stemmer Dictionary (AMESD)

The filtered words from the two languages are semantically unified through assigning common indices from the bilingual dictionaries.

The Arabic morphology, the English stemmer, the indexed bilingual dictionary and the data base access are all connected through our model called "MLTextMAES". The bilingual dictionary is retrieved through the MS Access driver in Windows XP (Data sources ODBC) using DBAccess. The execution process of MLTextMAES is composed of calling the Arabic Morphology and executing the English code stemmer to produce one single output file, read.txt. The read.txt file is served as an input to MLTextMAES which compares the text with the bilingual dictionary. Finally, Table 6.2 presents the indices which are generated after the comparison process.

6.2.6 Semantic Unification of Bilingual Dictionary

The roots in the bilingual dictionary are encoded in NENA structure (number-English number-Arabic). As each language has its own unique grammatical struc-

File Edit Format View Help

Izzet Ibrahim Meet Saudi Trade Official Baghdad Baghdad Iraq official new agency report D
Iraqi Revolute Command Council Izzet Ibrahim today AbdulRahman Zamil Manage Directo
Develop agency Ibrahim occasion trade exchange cooperate Iraq Saudi Arabia agency report
Trade part meet Baghdad Riyadh diplomatic relate Gulf War improve relate course Beirut S
source week deny report American NewYork Time Saudi Arabia agree allow United States S
base war against Iraq place Prince Abdulrahman Saudi Arabia Defense Minister state Saudi
newspaper incorrect add Kingdom posit clear able place airspace base American control Sau
base American force Gulf War
*

Israeli Army Destroy Palestinian Home Incursion Rafah Gazza official security source state
army destroy Palestinian home night incursion Rafah Palestinian Refugee south Gazza Strip
Press occupay force tank military mechan equip bulldozer penetrate meter Palestinian land B
Rafah Refugee complete destroy house damage house mid heavy gunfire Israeli soldier open
house force inhabit leave arm place arm Palestinian Israeli army information concern casua
Iraqi Oppose Prepare meet Kurdistan London Iraqi oppose group meet London prepare mee
consist member form London held Kurdistan member Committee France Press last night me
conference organize Iraqi oppos move London last course agreed meet take place Iraqi Kurdi
night agenda discuss Committee meet Iraqi Kurdistan define organize logistic prepar commi
Kurdistan prepar oppose schedule add agenda decide final decide concern conference place d
meet week
*

Kill Russian Traffic Accident Cairo new agency report people Russian kill injur traffic accid
Sinai last count add tourist coach carry tourist collide private car rent car Resort Sharm Shei
Sea Russian victim husband wife daughter young girl year man wife Egypt coach drive Britis

Figure 6.9: English Corpus After Stemming

Table 6.2: Example of Bilingual Dictionary

Index	Eng-Word	Ara-Word	Cat-No
10010001	accept	قبل	3
10020002	active	نشط	6
10030003	american	امريكي	4
10040004	art	فن	7
10050046	attend	حضر	3
10060006	attitude	سلك	6
10070007	average	معدل	2
10070008	average	متوسط	2
10080009	build	بنى	6
10120020	educate	درس	3
10310020	school	درس	3
10210020	learn	درس	3

ture we use "active" instead of words, and each root is given its own index (NENA) where NE stands for 4 digits English and NA for 4 digits Arabic.

We can start with the second word appearing in the English-Arabic dictionary, which is "active", meaning "نشط" in Arabic, and assign 1002 to "active" and 0002 to "نشط". Every new word in either English or Arabic is thus given a unique index. If any word is repeated due to multiple meaning, the index remains unchanged.

The numbers in both English and Arabic are then combined as in Table 6.2 below. The last field assigns a category number from 1-8 main categories.

6.2.7 Indices Generation

In order to check the functionality of Multilingual Morphological Analysis (MMA), we have selected M documents for Arabic and English as the test case. The MMA was then executed to obtain the final results. At the end of this stage, the results have been saved into single file "writedocs.txt", which will be the primary source for

the input data of the next stage. The array was generated by MMA of two datasets. It consists M columns and N rows of the relevant roots in both languages. Each column correspond a document either Arabic or English language. Hence, first row and first column is indexed by one byte, first part of the byte for English, while the second part of byte for Arabic. The method we followed in this study is believed to improve text mining in both languages. Table 6.3 shows the indices generated by MMA, for the set of Arabic-English corpus mentioned above. The table is generated by assigning index number to words occurring in the two sets of documents from the Arabic-English dictionary. For example, "65452468" in first column and first row represents the word "student" in English, while "65452468" in second column and third row means the word "طلب"in Arabic. These two documents are related to "education". The third document in English language is related to "sport" while the fourth document in Arabic language is again related to "education". The fifth English document related to "food". The six document is about "travel" in Arabic and the final document in English is related to "travel" as well. So we expect some common indices in the table below.

Table 6.3 presents the results achieved by MMA of two datasets. It consists seven columns and thirty nine rows of the relevant words in both languages. Each column correspond a document either Arabic or English language. Hence first row and first column is "65452468" in English related to "student", while "65452468" from first row and second column is presented to "طلب" in Arabic, then the first row and third column is "64002074" indicated to "sport" in English, while "38962845" related to "قدم" from first row and fourth column. Next the thirteenth row and fifth column "10144222" represented "اكل" in Arabic, while "74178124" in third row and sixth column represented "كون" in Arabic, second row and seventh column is "74178124" related "world" in English.

Table 6.3: Index Feature (Matrix)

	A	B	C	D	E	F	G
1	65452468	51112468	64002074	38962845	12142214	17362736	17362736
2	65457545	58052468	64007401	47062845	12140003	60036690	74174214
3	15102510	65452468	64007400	35702845	52974456	74176124	74176124
4	15102511	29282468	10782078	44892845	52973750	19952995	74178124
5	23043305	13712468	10782077	50922845	52976298	21262995	72832995
6	23043304	22942468	41705170	54692845	52972379	36312995	72836778
7	23043306	14682468	10062076	40022845	32224222	40722995	64596775
8	23040013	54692511	54042074	10662845	36984699	30464046	64596776
9	32484249	15112511	54043491	47185418	36984700	57794046	64597461
10	32484248	22892511	57616761	36005418	36984698	72496223	23183318
11	32484251	74562511	57610114	74935418	36980021	53556223	26793679
12	32485418	15102511	32484249	32485418	10140024	42285228	26795228
13	48615861	33823306	32484248	65595418	10144222	26795228	26790194
14	47185418	44653306	32484251	23663366	10143266	27362854	58876890
15	54692511	60413306	32485418	40033366	41374934	28392854	58876891
16	54692544	71313306	48885890	48043366	41372878	72062854	58872854
17	54694117	23043306	48885889	50383366	41373467	18542854	15042504
18	54692845	74703306	48885891	70033366	41372083	14772477	10000000
19	10382038	68203306	48882856	24490015	41375141	32894289	10000000
20	71298129	38224249	48882446	33823306	41375137	36424289	10000000
21	71293533	32484249	10260043	44653306	66507652	45534289	10000000
22	10000000	60074249	41705170	60413306	66507577	10000000	10000000
23	10000000	67864249	39084697	71313306	66500059	10000000	10000000
24	10000000	48625861	10000000	23043306	66502199	10000000	10000000
25	10000000	48615861	10000000	74703306	36984699	10000000	10000000
26	10000000	47185418	10000000	68203306	36984700	10000000	10000000
27	10000000	36005418	10000000	47185418	36984698	10000000	10000000
28	10000000	74935418	10000000	36005418	36980021	10000000	10000000
29	10000000	32485418	10000000	74935418	10140024	10000000	10000000
30	10000000	65595418	10000000	32485418	10144222	10000000	10000000
31	10000000	38962845	10000000	65595418	10143266	10000000	10000000
32	10000000	47062845	10000000	46450033	54383940	10000000	10000000
33	10000000	35702845	10000000	38224249	54380045	10000000	10000000
34	10000000	44892845	10000000	32484249	15482548	10000000	10000000

6.2.8 Corpus

SOM utilizes an unsupervised learning technique, and therefore needs a training corpus. Our experiments trained the system using Arabic-English documents collected from the Linguistic Data Consortium (LDC). They were mainly collected from the Agence France Presse (AFP) and the Xinhua News Agency [154]. The documents were categorized into sub-domains: politics, sports, culture, arts, science, technology, education, economics, health, medicine, geography, history and travel, etc. The text is mainly tagged with simple XML to mark different documents and paragraphs. In addition, we have used the International Corpus of Arabic (ICA) (2006) [155] from the School of Computing at Leeds University. The ICA includes 842684 words and 415 texts. The corpora was collected from website magazines and newspapers, which are XML marked-up files of the corpus.

6.3 Stage II Training the SOMMLTM Algorithm

In this stage we describe how the SOM neurons are interconnected to create a two-dimensional array, where each input is connected to a neuron via weight connections, in order to create a model of the neural network consisting of nodes (neurons) and weighted connections (synapses). Figure 6.10 below shows our network using the self-organizing map, consisting of usually an input layer, and an output layer. The input layer is the layer which is fed the data to be processed through the network, while the output layer visualizes the result of the network. The nodes in the different layers are connected by a series of connections, each assigned a different weight.

The learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too large, the algorithm may become unstable. Otherwise, the algorithm

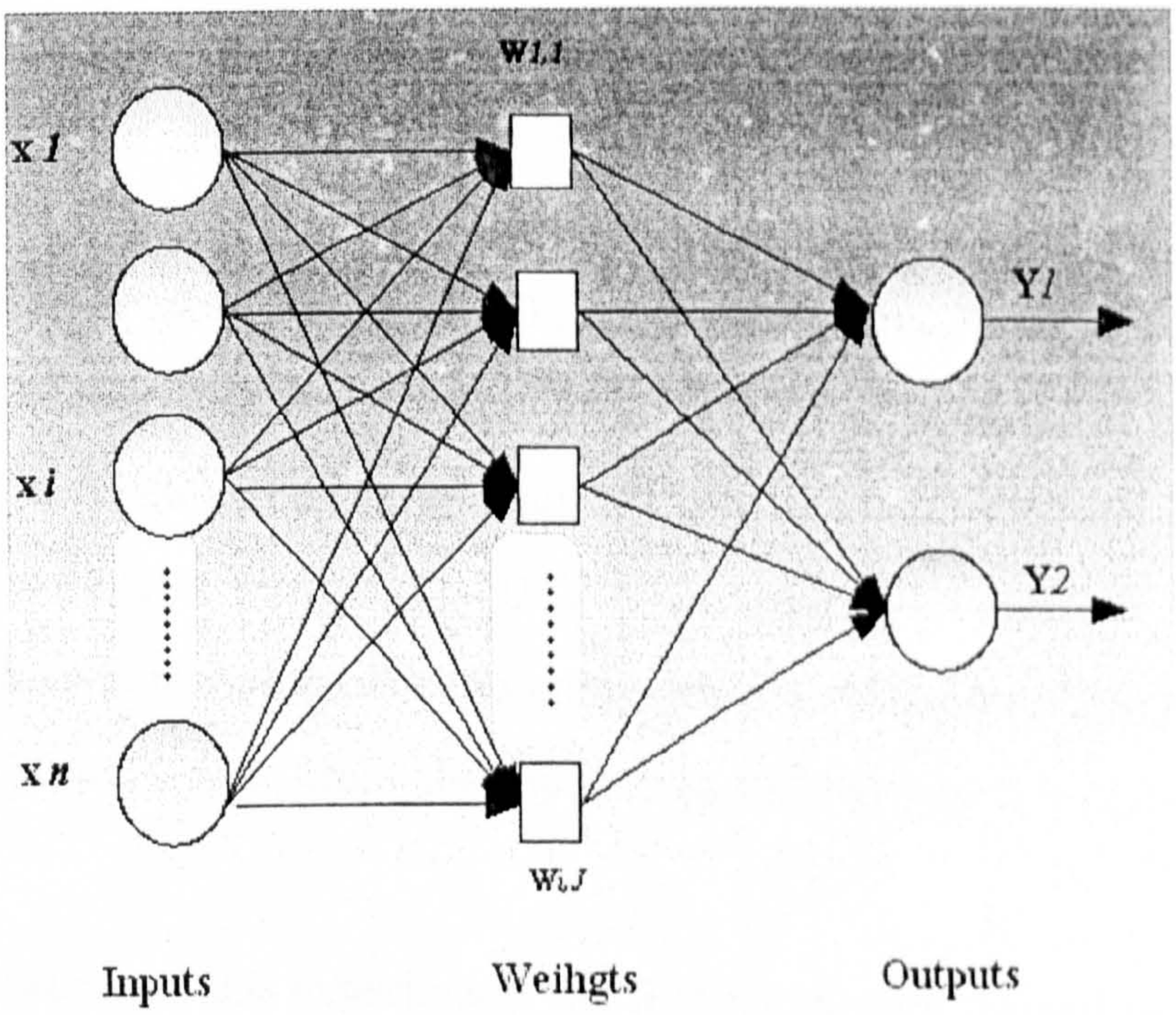


Figure 6.10: SOM Network

takes a long time to converge, because the learning rate is too small. It is not useful to determine the optimal setting for the learning rate before training. In fact, the optimal learning rate decreases during the training process, as the algorithm moves through the performance surface.

Training means the way by which multi-dimensional data is reduced to lower-dimensional spaces, usually to one or two dimensions. With the Kohonen technique, a network is created to store information, maintaining any topological relationship established during the training [2].

6.4 Constructing of the Maps

In order to evaluate the experimental results in Arabic and English, we have selected a sub-dataset of 40 documents from the 8 main categories in the bilingual classified dictionary using the criterion that they could be classified easily by themselves i.e., we know their exact meaning. We also added several countries and city names,

which were not in the dictionary but appear frequently in the documents. All source data were selected from the two newswire, the story were collected from January and February 2003 from Xinhua and AFP Arabic news data [148]. All selected stories contain between 700 and 1500 Arabic characters. For the Arabic data, there are approximately 15K words, while 135k words for the English translations (see a sample of a transcription in appendix C); in total and 10K unique words. The source files were later converted to UTF-8 encoding (8-bit Unicode Transformation Format), which it makes it easier to apply pre-processing for these documents. We used an SOM network to train the input data ASMA, which is the outcome of the first stage.

During the course of this experiment, the first maps were trained using parameters which has been selected according to the guidelines presented in Section 6.3. The best maps, rated according to quantization error (Equation 3.5) and ease of readability, were then selected and used as a basis when training further maps. The best map is indicated by the smallest quantization error. Table 6.4 below represents a map grid, sized 6X4 nodes, which integrates the data for 40 documents.

At the final phase, a number of steps, were generated directly from the recommendations provided above. The initial phase includes 1000 steps. The learning rate factor was set to 0.5 in the first phase, very near the recommended starting point. The neighborhood radius was set to 7 for the first phase. Compared to the recommendations, the initial radius was very large but seemed to provide the overall best map.

As Kohonen [2] noted, the selection of parameters appears to make little difference in the outcome when training small maps. As long as the initial selected parameters remained near the guidelines presented above, the changes in the quantization error were small. Table 6.4 illustrated some example of the outcomes and the parameters. The differences are small in the results, these are only a fraction of the entire

training set.

In this case, these capabilities of SOM have been used to cluster the map. Document clustering has been used for this purpose as this greatly increases the objectivity and accuracy of the cluster analysis. As was mentioned above, the maps were trained by using the entire dataset, from 2003-2005, to create a single map.

6.5 Stage III (Quality of Test and Data Visualization)

To assess the MLTextMAES model once control measures have been implemented, the improved Arabic-English should be assessed to determine if data quality standards have been attained. The collecting software used to gather the data for this assessment should be based on the specific languages problems or sources.

6.5.1 Quality of Test

We have tested our developed model on a single data set composed of 40 documents from Arabic-English corpus. Size of output map in each test is $7X5$. Results are compiled in Table 6.4. Training is performed in two stages. In stage-I, large initial radius and learning rates are used. Tests are performed with varying initial rates and the number of generated clusters is given. In stage-II small neighborhood radius and initial learning rate are set. Quantization error is the measure of our results. It is to be noted that our developed model is consistent in reproducing almost the same q.e. with different initial learning rates in stage-I of training. Number of iterations varies in different tests despite the same parameters used is due to the reason that the initial weight matrices are randomly generated in each run. This shows the random initialization process of the SOM, but also proves that the results from one map to another are consistent. Depending on the training SOM stage and using

the output of the SOM network, the data is visualized in the form of U-matrix. We can now show that the output of this model consists of clusters of all documents for Arabic-English text documents into a two-dimension map as shown in Figure 6.11 below.

Table 6.4: Example of Trained Maps

First Training Stage					Second Training Stage				
Maps	*Documents*	*Radius*	L-r	*Iterations*	*Radius*	L-r	*Iterations*	*Q.e.*	*Clusters*
Map1	40	7	0.9	1250	1	.05	5000	32.500	2
Map2	40	7	0.7	500	1	.05	2000	32.406	3
Map3	40	7	0.6	500	1	.05	2000	33.205	3
Map4	40	7	0.5	1250	1	.05	5000	32.080	6
Map5	40	7	0.1	1250	1	.05	5000	32.907	4

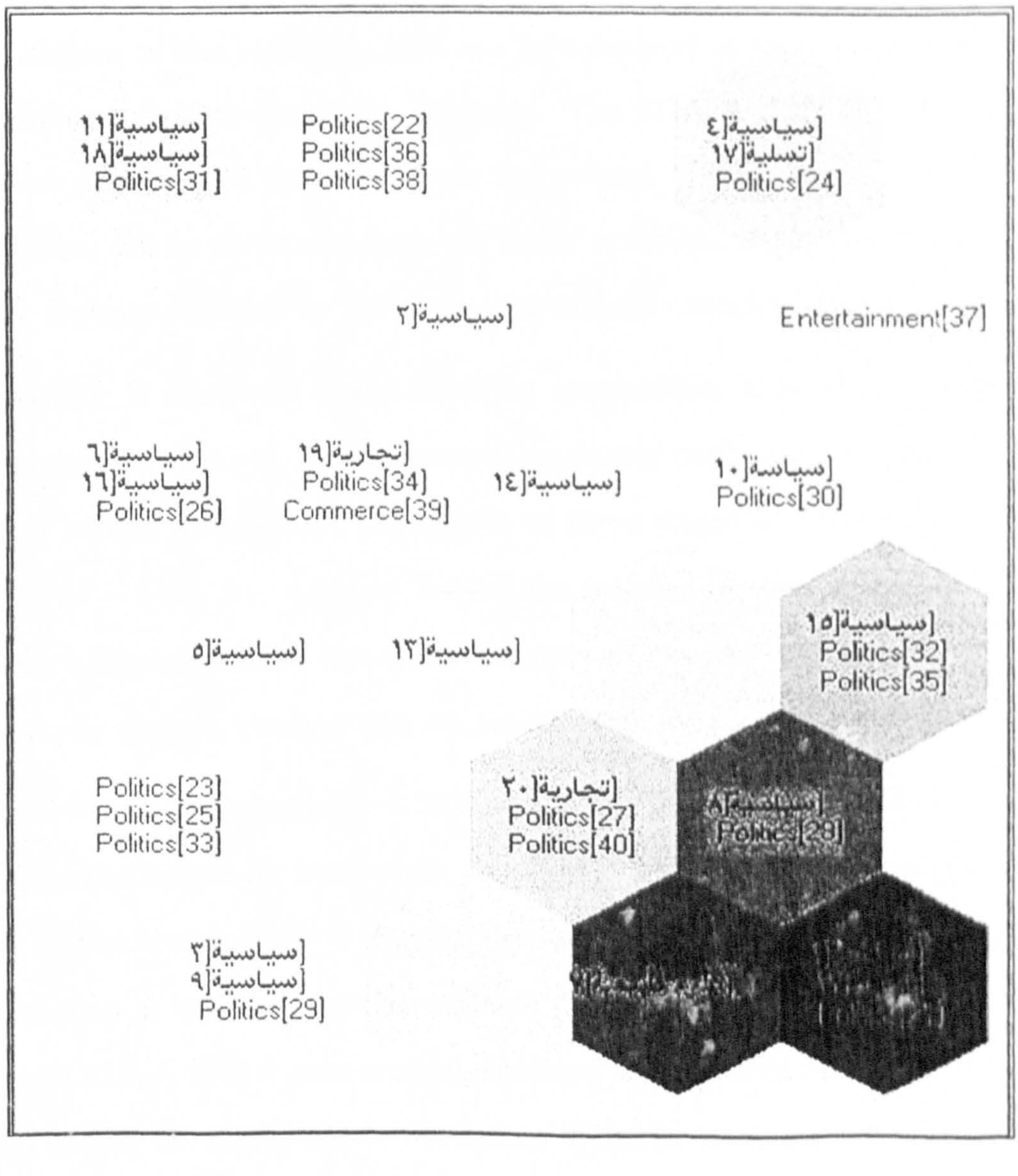


Figure 6.11: Visualization of the Similarity of 40 Arabic-English Documents from LDC

6.6 Summary

The implementation of our MLTextMAES model involves three stages: filtering and stemming, numeric matrix generation, and training and construction. The current implementation of the MLTextMAES can be enhanced in several ways by changes in the implementing programming language. The $SOM - DOCS - ASMA$ function enable this mission flexibility with the current Matlab implementation. The entire mission can be performed more efficiently and without need for complex error-handling routines required by the sequential cyclical executive approach.

Pre-processing is composed of the following components: stop words removal, prefixes and suffixes removal, infixes removal, semantic unification of bilingual dictionary, and indices generation. The output of these stages is input for the training of the $SOM - DOCS - ASMA$. Within the training process, a map is built and the network organizes itself through a competitive process. New input vectors are automatically given a location and are classified, finding a single winner neuron.

The operation of the algorithm is in three steps: the corpus is analysed in order to generate the indices for every root; the structure dataset, ASMA, is constructed; and the output is visualized on the grid through training the SOMMLTM algorithm. The best map is indicated by the smallest quantization error. Assessment of the $SOM - DOCS - ASMA$ once control measures have been implemented for Arabic-English, should be conducted to determine if cluster quality standards have been attained. For example, past experience has shown that when English-Chinese are arrayed in grid to detect trends, cluster quality is improved.

As shown in Figure 6.11, a MLTextMAES that allocates data to meet cluster quality standards must be established. The biggest advantage of the SOM algorithm in text data mining applications, is its efficiency in clustering large data sets (see the comparison with feed-forward backpropagation technique in Section 3.2.2) . How-

ever, its use is limited to numerical values. The SOMMLTM algorithm presented in chapter 5 has removed this limitation whilst preserving its efficiency.

Chapter 7

Evaluations

Objectives

- To present the corpus encoding
- To present the results of experiments
- Evaluation and discussion

7.1 Introduction

In order to assess the various computational models of similarity, we have gathered some subjective human ratings of similarity with respect to text documents; these assign categories to text documents according to their similarities. In this work, we have measured these classifications for document similarity as determined by human experts and MLTextMAES. We performed several experiments on our model of monolingual and multilingual text mining. Typically there are two types of corpus, derived from either written or spoken language: raw corpus and annotated corpus. The former mainly features the text itself with no other additional information, and

the latter is text which has been noticeably enriched with a variety of information. Each corpus is used here to measure the software's performance.

7.2 Corpus

In testing our system, we have used two different corpus. As we will shed some light on data sets used to conduct our experiments. The major corpus we have used are:

1. LDC (Linguistic Data Consortium).

2. ICA (International Corpus of Arabic).

The following table presents the summaries of the total corpus.

Table 7.1: Summary of Sources

Corpus	Type	Number of Files	Number of Words
LDC (source)	Newswire	100	15,056
LDC (translation)	Newswire	700[1]	101,857
ICA	website magazines and newspapers	427	842,684
Total		1227	959,597

7.2.1 Linguistic Data Consortium Corpus (LDC)

The corpus used in this study was developed by Linguistic Data Consortium (LDC) (2003) [148], at the University of Pennsylvania. Its aim was to provide a resource for education and for the development of technology. The text is mainly tagged with simple XML to mark different documents and paragraphs.

LDC corpus consists of multiple translations of text from one source language, Multiple-Translation Arabic (MTA) Part 2. Part 2 is about corpus composed of 100 Arabic documents see a transcribed sample Arabic document in Figure 7.1 below.

Additionally for each Arabic document of the corpus 7 different English translations. Arabic corpus translated with four human translations ahd, ahe, ahg, ahi, see a translation sample of this data below and three different translations corpus in English by machine system id's are ama, ame, and arp (see the details in Table 7.2 and Table 7.3). Its purpose is to investigate how the lexis of information technology and lexical collocations are translated into English see a transcribed sample of this data below.

Each English corpus contains around 14K words and the Arabic corpus contains 15K words. The texts are mainly selected from the two newswires and the length of each text is between 700 and 1500 Arabic characters. The texts are collected from the Agence France Presse (AFP), and Xinhua News Agency. This corpus is not available for public use unless the copyright permission is obtained for academic research investigations.

File Edit Format View Help

<DOC docid="AFA20030101.5900"><hl><seg id=1> عزة ابراهيم يستقبل
-(بغداد 1-1 (اف ب <seg id=2></seg></hl> مسؤولا اقتصاديا سعوديا في بغداد
ذكرت وكالة الانباء العراقية الرسمية ان نائب رئيس مجلس قيادة الثورة في العراق عزة
ابراهيم استقبل اليوم الاربعاء في بغداد رئيس مجلس ادارة المركز السعودي لتطوير
وقالت الوكالة ان <seg id=3> .الصادرات عبد الرحمن الزامل</seg>
.ابراهيم رحب في المناسبة بمستوى التعاون والتبادل التجاري بين العراق والسعودية
واشارت الوكالة الى ان وزير التجارة العراقي محمد <seg id=4></seg>
وبدأت بغداد والرياض <seg id=5> .مهدي صالح شارك في اللقاء</seg>
اللتان قطعتا علاقاتهما الدبلوماسية منذ حرب الخليج في 1991، عملية تقارب خلال قمة
ونفت السعودية هذا <seg id=6> .بيروت في آذار/مارس الماضي</seg>
الاسبوع معلومات نشرت في صحيفة "نيويورك تايمز" الاميركية وفيها ان السعودية وافقت
على وضع منشآتها العسكرية في تصرف الولايات المتحدة، في حال حصول حرب مع
وقال الامير عبد الرحمن بن عبد العزيز نائب <seg id=7> .العراق</seg>
وزير الدفاع السعودي في تصريح لصحيفة "عكاظ" السعودية ان "ما ادعته الصحيفة كلام
واضاف ان "موقف المملكة واضح من <seg id=8> ."غير صحيح</seg>
.البداية تجاه هذا الامر ونحن لا يمكن ان نضع مجالنا الجوي وقواعدنا بتصرف" الاميركيين
وشكلت السعودية قاعدة للقوات الاميركية خلال حرب <seg id=9></seg>
.الخليج (كانون الثاني/يناير وشباط/فبراير 1991
</seg> </DOC>

Figure 7.1: The Original Arabic Document

Human Translation Procedure

To support the development of automatic means for evaluating translation quality, the LDC was sponsored to solicit 4 sets of human translations for a single set of Arabic source materials. System performances are evaluated on LDC distributed multiple translation Arabic Part2 consisting of 116913 words obtained from AFP and Xinhua newswires. Translation qualities are measured by uncased BLUE [156] with 4 reference translations ahd, ahe, ahg and ahi, see a sample of Human ahd scheme for English document in Figure 7.2. Table 7.2 shows the translation procedure adapted by the 4 human scheme.

Table 7.2: Human Translation Team Information

Team ID	Software Used	Translation Procedure
ahd	No	The translator reads the text through to grasp the gist of it and identify the register. He then starts translating it, using all the usual translation techniques of modulation, transposition, etc. The story is reread, and problematic names of people, organisations or places, are researched; any relevant changes are made. Finally, after finishing a whole batch, the translation is checked for spelling errors and re-read to make sure there are no omissions. The translation is then sent to the proofreader. The proofreader first read the entire translation, marking anything that looked inconsistent or questionable, and correcting any spelling or grammar mistakes. Then the translation is compared to the source for accuracy. This step also involved researching items such as accepted spelling of names, English equivalents to Arabic organizations, etc. Finally the entire translation is read one or two more times, making any necessary changes.
ahe	No	The translator prints out the Arabic text and from that performs an initial draft of the translation. He then revises the draft, looking up any proper names in encyclopedias, atlases and the internet. The spelling is checked using the Word for Windows spell checker. The proofreader calls up the Arabic text on one half of the screen and the English on the other. The English is then amended as appropriate, a record is made and the amended text and record is released.
ahg	No	Prints of the source text taken, text translated directly into the computer, followed by a revision by the same translator. It is then cross-checked by an English editor/project manager for grammatical aspect, layout, consistency etc. Wherever required, help is taken from the alternate translator.
ahi	Yes	Translator uses Al-wafi software tool and does the translation and then passes it to the proofreader for proofreading. The translations are checked by a third person (usually the project manager) to ensure that formatting and other issues are in accordance with the client's instructions. The translator and proofreader will confer about significant changes or where understanding is incomplete.

File Edit Format View Help

<DOC docid="AFA20030101.5900" sysid="ahd"><hl><seg id=1>
Izzet Ibrahim Meets Saudi Trade Official in Baghdad </seg></hl> <seg id=2> Baghdad 1-1 (AFP) -
Iraq's official news agency reported that the Deputy Chairman of the Iraqi Revolutionary Command Council, Izzet Ibrahim, today met with Abdul Rahman al-Zamil, Managing Director of the Saudi Center for Export Development. </seg> <seg id=3>
The agency said Ibrahim welcomed this occasion for trade exchange and cooperation between Iraq and Saudi Arabia. </seg> <seg id=4>
The agency also reported that the Iraqi Minister of Trade, Mohamed Mehdi Salih, took part in the meeting. </seg> <seg id=5>
Baghdad and Riyadh, who broke diplomatic relations during the Gulf War in 1991, began to improve their relations over the course of the Beirut Summit last March. </seg> <seg id=6> Saudi sources this week denied reports in the American New York Times that Saudi Arabia had agreed to allow the United States to use Saudi military bases should a war against Iraq take place. </seg>
<seg id=7> Prince Abdulrahman bin Azziz, Saudi Arabia's Defense Minister, said in a statement to Saudi newspaper Uqath that "what was claimed by the newspaper is incorrect." </seg> <seg id=8> He added: "The Kingdom's position has been clear since the beginning and we are not able to place our airspace and bases" under American control.
</seg> <seg id=9> Saudi Arabia was a base for American forces during the Gulf War (January and February 1991). </seg> </DOC>

Figure 7.2: Human ahd scheme for English Translation

Machine Translation (MT) System Information

Complete sets of automatic MT were also produced by submitting the 100 files to each of the two MSRLM: a scalable language modeling toolkit [157] and ATA Machine Translation Software [158] publicly-available MT systems Commercial-Off-The-Shelf (COTS system) and including commercial Machine Translation (MT) systems as well as MT systems available on the Internet (http://research.microsoft.com/ research/ downloads /details/ 78e26f9c-fc9a-44bb-80a7-69324c62df8c/details.aspx). Table 7.3 shows the Arabic corpus translated into three different translations corpus in English by machine (System ID's are ama, ame, and arp) see a sample of machine ama scheme translation for English document in Figure 7.3.

Table 7.3: Machine Translation Procedure

System ID	System Information	Translation Procedure
ama	Commercial software	Built-in batch translation was used. Unrecognized Arabic words are preserved in the output. Thus output is a mix of plain ASCII and CP1256, which is converted into UTF8 later.
ame	Commercial software	No details available.
arp	Research system	No details available.

File Edit Format View Help

<DOC docid="AFA20030101.5900"><hl><seg id=1> عزة ابراهيم يستقبل
-(بغداد 1-1 (اف ب <seg id=2> مسؤولا اقتصاديا سعوديا في بغداد </seg></hl> <seg id=2>
ذكرت وكالة الانباء العراقية الرسمية ان نائب رئيس مجلس قيادة الثورة في العراق عزة
ابراهيم استقبل اليوم الاربعاء في بغداد رئيس مجلس ادارة المركز السعودي لتطوير
وقالت الوكالة ان <seg id=3> .الصادرات عبد الرحمن الزامل </seg> <seg id=3>
.ابراهيم رحب في المناسبة بمستوى التعاون والتبادل التجاري بين العراق والسعودية
واشارت الوكالة الى ان وزير التجارة العراقي محمد <seg id=4> </seg> <seg id=4>
وبدأت بغداد والرياض <seg id=5> .مهدي صالح شارك في اللقاء </seg> <seg id=5>
اللتان قطعتا علاقاتهما الدبلوماسية منذ حرب الخليج في 1991، عملية تقارب خلال قمة
ونفت السعودية هذا <seg id=6> .بيروت في آذار/مارس الماضي </seg> <seg id=6>
الاسبوع معلومات نشرت في صحيفة "نيويورك تايمز" الاميركية وفيها ان السعودية وافقت
على وضع منشآتها العسكرية في تصرف الولايات المتحدة، في حال حصول حرب مع
وقال الامير عبد الرحمن بن عبد العزيز نائب <seg id=7> .العراق </seg> <seg id=7>
وزير الدفاع السعودي في تصريح لصحيفة "عكاظ" السعودية ان "ما ادعته الصحيفة كلام
واضاف ان "موقف المملكة واضح من <seg id=8> ."غير صحيح </seg> <seg id=8>
.البداية تجاه هذا الامر ونحن لا يمكن ان نضع مجالنا الجوي وقواعدنا بتصرف" الاميركيين
وشكلت السعودية قاعدة للقوات الاميركية خلال حرب <seg id=9> </seg> <seg id=9>
.الخليج (كانون الثاني/يناير وشباط/فبراير 1991
</seg> </DOC>

Figure 7.3: The Machine ama Scheme Translation for English Document

In this model, all of the training data that have been used were extracted from the Linguistic Data Consortium (LDC) and the International Corpus of Arabic (ICA). The experimental design, choices of scheme and techniques were selected independently of the test set. The datasets (LDC2005T05) include one Arabic source and seven English reference translations.

7.2.2 International Corpus of Arabic (ICA)

The second corpus we have used for testing our system is International Corpus of Arabic (ICA) (2006) [155] from the School of Computing at Leeds University. ICA contains 14 small corpus in various fields (Economics, Sports, Religion, Children's Stories, Science, Recipes, ScienceB, Short Stories, Politics, Tourist and Travel, Sociology, Health and Medicine, Interviews and Autobiography). The ICA includes 842684 words and 427 texts. The corpora was collected from website magazines and newspapers, which are XML marked-up files. Transcribed samples of these data are given in Figure 7.4. The International Corpus of Arabic (ICA) is available freely for the public from the website.

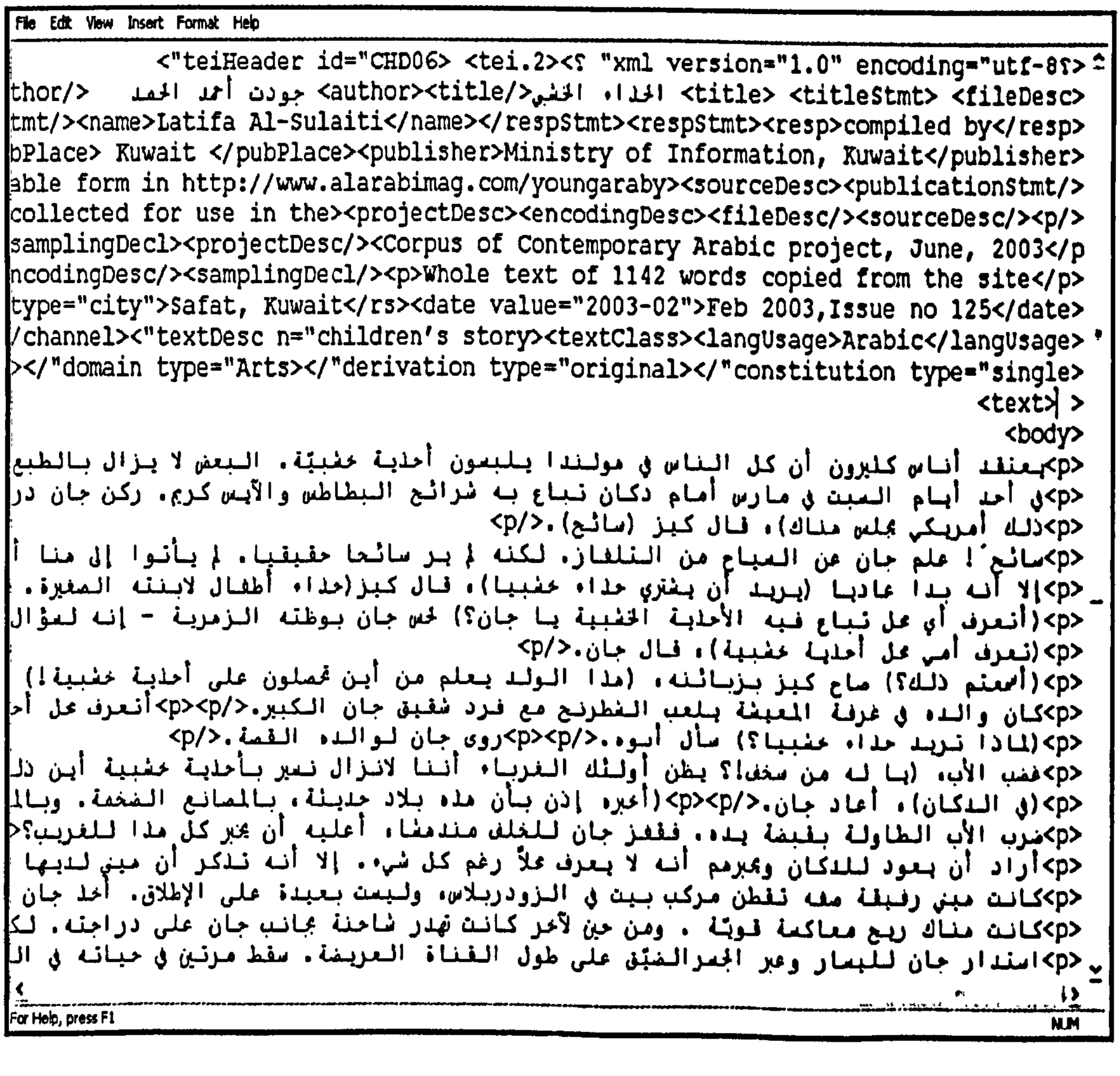


Figure 7.4: Transcribed sample of ICA Corpora

7.3 Procedure For Encoding Corpus

Encoding large corpora is usually done by computer programs so that many texts can be encoded in a short time. This is normally done automatically. Below is an explanation of the main procedures for encoding a corpus:

1. Select a text from the Internet or other system and save it as encoded text choosing Unicode UTF-8.

2. removing elements not needed to be included in the corpus and save it.

3. Rename the text by changing the file extension from .html to .txt.

7.4 Experiments and Results

We have performed five experiments. Experiment 1 and 2 are preformed on monolingual. Experiment 3, 4, and 5 are performed on multilingual Arabic-English. The analysis of our results are given in Section 7.9.

7.4.1 Experiment 1

Objective and Setup

This experiment is conducted on selected 20 Arabic documents with no English translation to test our system on monolingual documents. All the 20 documents taken from LDC comes under the field titles of Political, Social Science, commerce, Life and Leisure.

Our MLTextMAES was applied to monolingual corpus, in order to determine whether or not our developed model is satisfactorily effective in dealing with monolingual text mining. The sub-corpus contained 20 documents of Arabic texts.

Properation

To reduce the dimensionality, we applied pre-processing to remove stop words, punctuation and html tags. This also reduced the size of the documents, and reduced the memory spaces. We then constructed an SOM that consisted of 20 neurons in a 4X3 grid format. The initial iteration time was set to 1000 and the α=0.5. **Results**

Figure 7.5 illustrates the results of the documents and the respective clusters obtained via training with SOMMLTM algorithm. Clusters with the lightest shades are strongly correlated (5, 14, 19, 4, 11, 18). While, the darkest shade represents the weak correlation between the documents (eg. 1, 2, 7, 12).

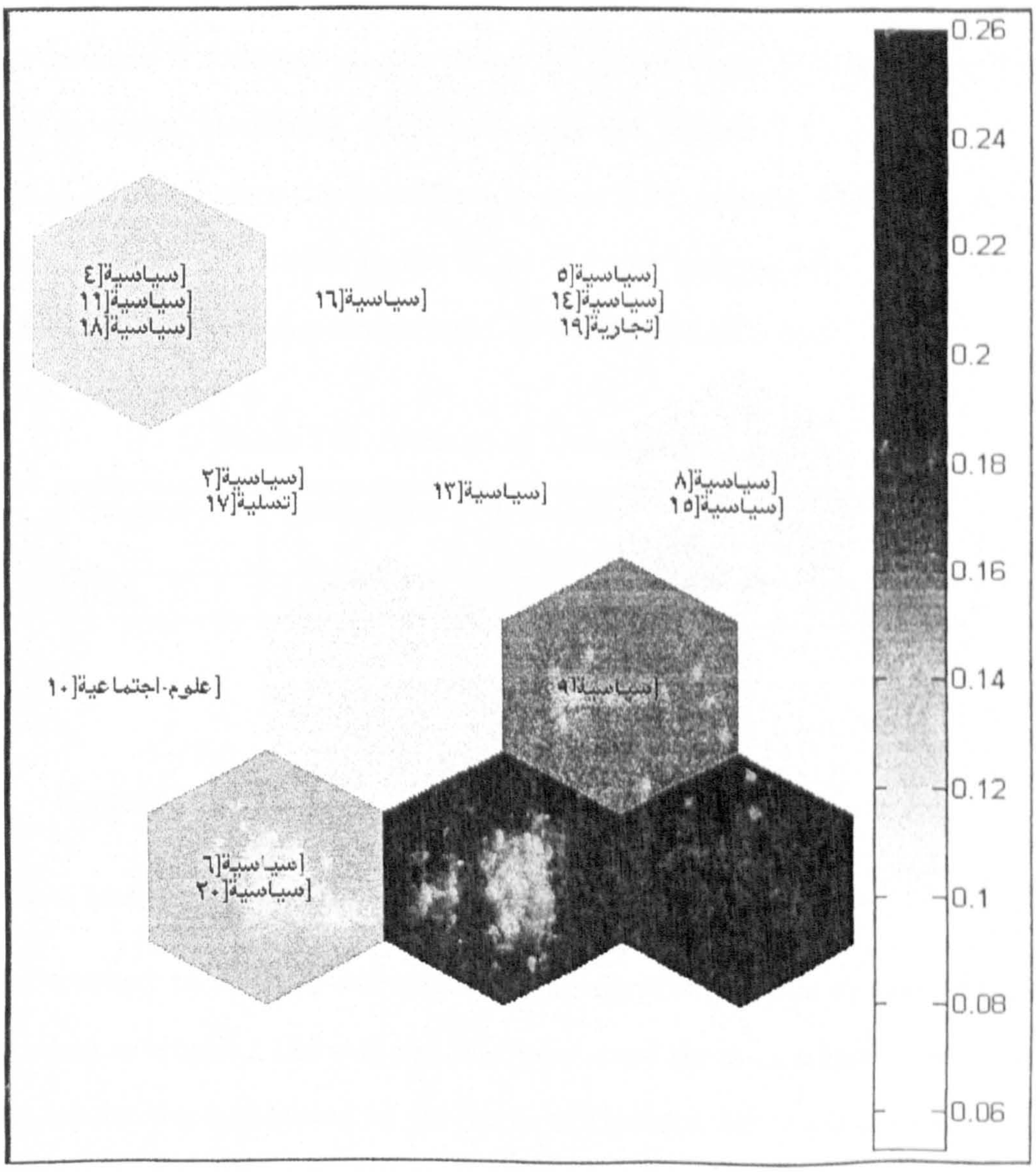


Figure 7.5: Training of Monolingual Sets

7.4.2 Experiment 2

Objective and Properation

Essentially this experiment is similar to experiment 1 perform earlier but on large monolingual (Arabic dataset). The corpus is obtained from the freely available International Corpus of Arabic [155]. The corpus is mainly related to (Political, Social Science, commerce, Life, Apply Science and Leisure). The size of the corpus is 427 Arabic documents with 842,684 words.

Results

Same procedure is followed as was done for experiment 1 that is preprocessing, filtering, indexing, stemming, SOM training, etc. Table 7.4 compiles the output obtained. Column 2 gives the quantization error 3.92, column 3 gives the percentage of keywords ultimately utilize in the SOM training process after the preprocessing stage. Also the total cputime consumed by MLTextMAES code.

Table 7.4: Average of Quantization Error

Scheme	Quan-Error	Words Percentage	Cputime Sec.
ICA	3.93	60%	2820.353

7.4.3 Experiment 3

Objective and Properation

We have selected 16 random documents from the main corpus to evaluate the clusters obtained in training the sample, and have used them as a test case. In general, the final results were obtained in the form of clusters, i.e. each document was assigned into one of the 8 main domains.

Results

As the clustering results were slightly different with each run, due to the randomly initialization of the weight matrix in SOM algorithm, the experiments were repeated 470 times (a snapshot of the 470 trials is given in Figure 7.6 below. This was done to gain an accurate statistical analysis. Normal distribution was selected for the analysis. Precision is usually characterised in terms of the standard deviation of the measurements. The interval, defined by one standard deviation away from the mean on either side, is 68.3% (1 (σ)) of confidence in terms of the measurements [159]. In our case, we may say that it is likely that 68.3% of the time, the mean value will lie within one standard deviation, or 95% of the time, within (2 (σ)). The mean (μ) and standard deviation (σ) turn out to be (μ=2.01915) and (σ= 1.49828) respectively. Figure 7.7 shows the resultant normal distribution for our experiment.

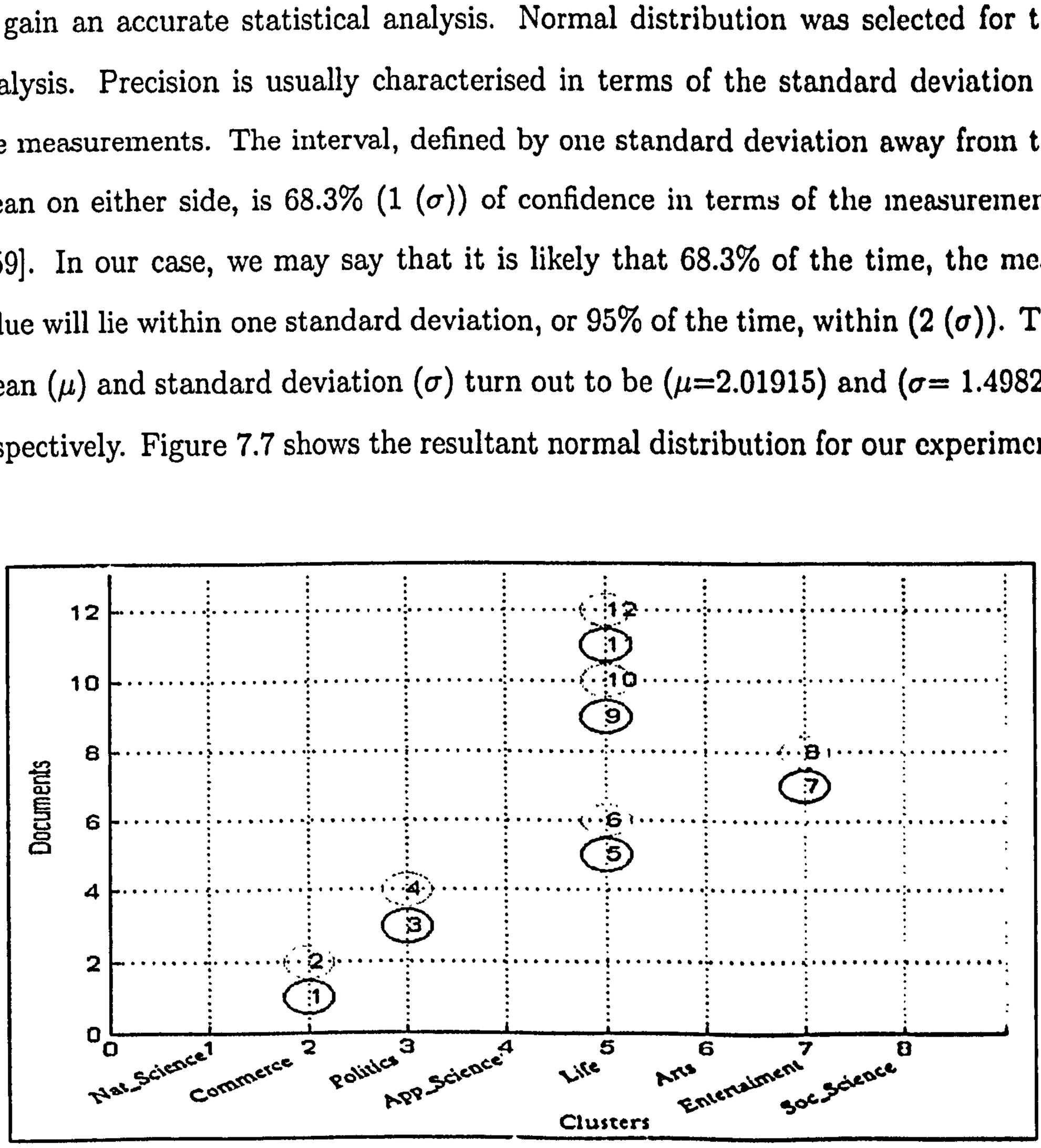


Figure 7.6: Screen Shot of One of the Trials

In order to evaluate our MLTextMAES model, we have selected eight Arabic and eight English documents as the test case. The MLTextMAES was then executed to get the final result. To illustrate, we began with the selected documents both in English and Arabic, and the results are shown in Figure 7.6.

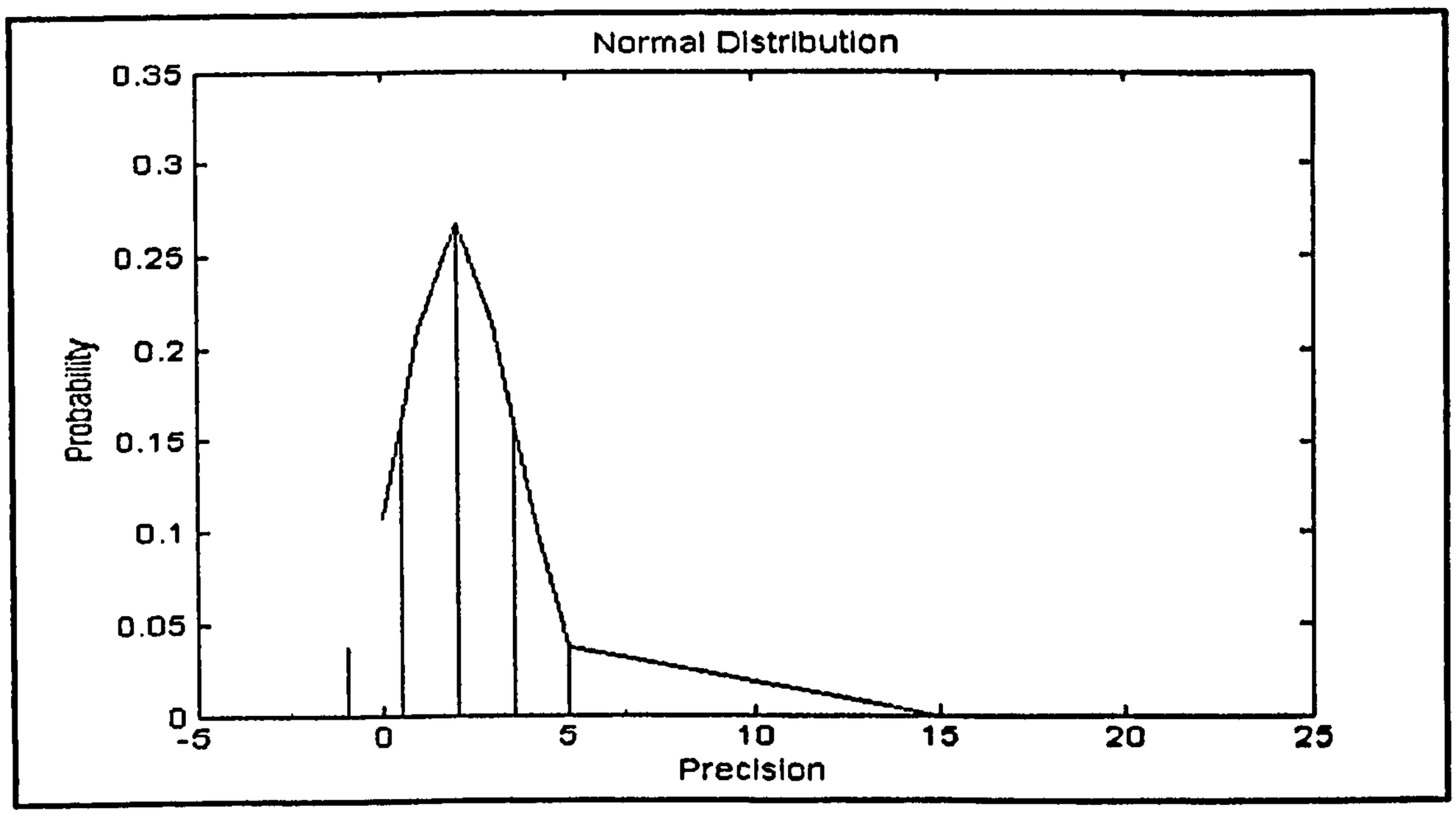


Figure 7.7: Standard Deviation

Figure 7.6 above shows the set of sixteen documents mentioned. The Arabic documents correspond to the green circle and the English correspond to the red one.

Among the eight main cluster domains, only five are occupied in this test case, viz.:

- Commerce
- Life
- Politics
- Social Sciences
- Entertainment

The results, i.e. the documents clustered together, are correct and this is due to the fact that the documents used in this trial are exact translations of the each other.

7.4.4 Experiment 4

Objective and Setup

This experiment is perform to test our model MLTextMAES with bilingual corpus. Twenty Arabic documents are randomly selected from LDC source, and there respective twenty English translations [148]. Fields related to the bilingual (Arabic-English) varies from applied Science to Political to arts and leisures. MLTextMAES is applied in bilingual mode on the selected LDC corpus.

Properation

We have manually selected 20 Arabic and 20 English documents from the 8 main domains in the bilingual dictionary using the criterion that they could be classified clearly [147].

At this stage, after applying the same pre-processor used for the training phase, the input of this part is unlabeled Arabic-English documents. The output is clustered documents in one of the categories. Due to the scarcity of high quality data of documents containing roots only, we gave our students a copy of the test documents to find out how humans assess the categories for these documents. This required expertise and an ability to effectively evaluate the subject matter. Therefore, in this experiment our model produced clusters for 2 corpus.

Table 7.5 shows the number of texts derived from each sub-category, which are then displayed in one of the eight main domains.

Table 7.5: Number of Texts in Each Sub-Category

Code	Text Category	Number of Texts
1	Education, تربية	n/a
2	Health, Medical, صحة, طب	n/a
3	Financial documents, وثاءيق مَالية	n/a
5	Geography, جغرَافيَا	n/a
6	Television, التلفزيون	n/a
7	Sports, ريَاضة	n/a
8	Newspapers, الصحف	n/a
9	Religion, دين	2
10	History, تَاريخ	2
11	Law, قَانون	2
12	Arts, فنون	n/a
13	Academic papers, الأورَاق العلمية	n/a
14	Advertisement, الاعلَانَات	2
15	Cultural, ثقَافة	2
15	Conversation, محَادثه	2
16	Plays, الألعَاب	n/a
17	Weather, طقس	n/a
18	Entertainment, الترفيهية	2
19	Tourist-Travel, سيَاحة وسفر	n/a
20	Transport, موَاصلَات	n/a
21	Technology, تقنية	n/a
22	Restaurants, مطَاعم	n/a
23	Economics, الاقتصَاد	2
24	Media, الاعلَام	n/a
25	Inner, سيَاسة دَاخلية	2
26	Military, الجيش	4
27	Wars, الحروب	6
28	World, العَالم	6
29	Government, الحكومة	2
30	Weapons, الأسلحة	4

Results

The Figure 7.8 presents a summary of the classification of the documents in each domain, where 40 documents were divided into eight categories (see Section 5.2.14). Also for the seek of a valuation of our developed model, human classification of these documents was also performed.

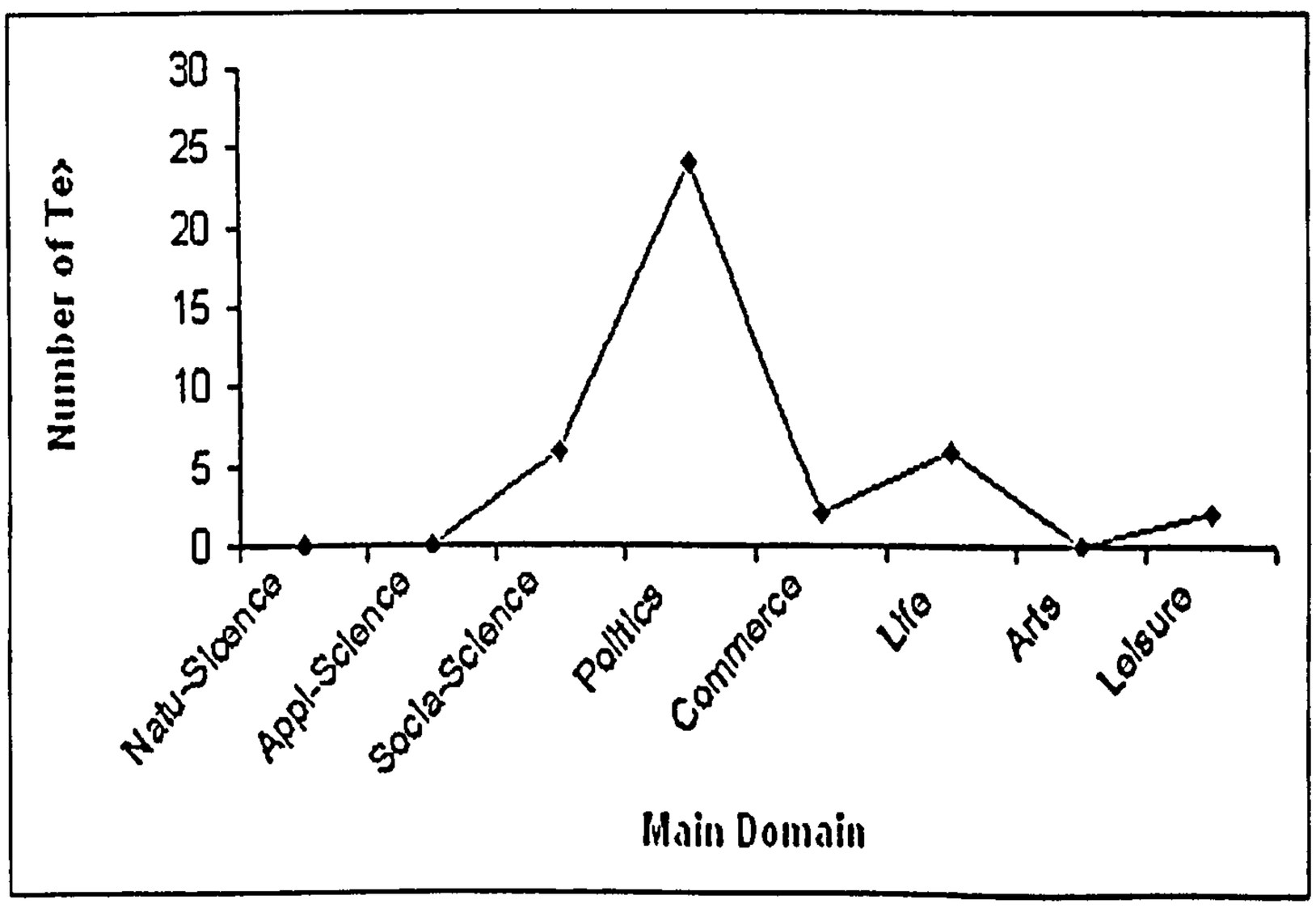


Figure 7.8: Number of Documents in each Domain

Table 7.6 and Table 7.7, if we compare these classification results with the map obtained through MLTextMAE, we find that the results are very similar. SOM is slightly better than that obtained with human decisions. However, SOM incorrectly classify seven documents, in the sense that these seven documents are only differently labeled with respect to the categorization in the dictionary (see Tables 7.5). Next we compared the clustering results from human classification and machine (MLTextMAES) classification. We distributed 40 documents (20 Arabic and 20 English) to a number of students to study and classify these documents. The results obtained are shown in Table 7.6. According to this, around 85% of the results are

correct by human classification. The MLTextMAES model was then applied to the same set of documents given earlier to the students, and the results are compiled in Table 7.7. It can be clearly observed that both human and machine classifications are quite close to each other.

Neither human nor MLTextMAES classification is 100% accurate. Humans do differ in categorizing and their decisions also depend upon their backgrounds. It is expected that less educated persons are more likely to make wrong decisions in this respect.

All values are expressed as standard error mean (S.E.M) of measurements of at lease three different experiments with three replicates in each. Analysis of variance (ANOVA) with the Benferroni Comparison criterion was done for all pairs of columns for the comparison of group means. The level for a significant difference was set at $P < 0.05$. All statistical analysis were performed using GraphPad Prism (GraphPad software).

Table 7.6: Comparison of Human Performance on Documents Classification

Doc.no	1	2	3	4	5	6	7	8	9	10
Code	Po	Po	Po	Li	Po	Po	Po	Po	Po	Po
Human-1 English	4	4	4	6	4	4	1	1	4	4
Human -2	4	4	4	6	4	4	4	4	4	4
Human -3	4	4	4	6	4	4	4	4	4	4
Percentage Average	100%	100%	100%	100%	100%	100%	66%	66%	100%	100%
Human -4 A abic	4	4	4	3	4	4	1	4	4	4
Human -5	5	4	4	3	4	4	4	4	4	4
Human -6	5	4	4	6	4	4	4	4	4	4
Percentage Average	33%	100%	100%	33%	100%	100%	66%	100%	100%	100%

11	12	13	14	15	16	17	18	19	20	Percentage
Po	Li	Po	Po	Ss	Po	Le	Po	Co	Po	
3	4	4	4	3	4	8	4	5	4	80%
6	4	4	4	3	4	8	4	5	4	90%
6	4	4	4	4	4	8	4	5	4	85%
0%	0%	100%	100%	66%	100%	100%	100%	100%	100%	
Avcrage value of Correctness										85%
4	4	4	4	4	4	8	4	5	4	80%
4	4	4	4	4	4	8	4	5	4	80%
4	4	4	4	3	4	8	4	5	4	90%
100%	0%	100%	100%	33%	100%	100%	100%	100%	100%	
Average value of Correctness										84%

Then we have tested the human performance and the MLTextMAES model. The results of this analysis are demonstrated in the following tables for Arabic and English. When a human takes a subtly different decision on classification, it has consequences for the categorization of certain documents, placing them in different domains (Figure 7.8). Although, the code of the categories are mentioned in Table 5.1.

Table 7.7: Comparison of MLTextMAES Performance on Documents Classification

Doc.no	1	2	3	4	5	6	7	8	9	10
Code	Po	Po	Po	Li	Po	Po	Po	Po	Po	Po
Exp-1 English	4	4	4	4	4	4	3	4	4	4
Exp-2	4	4	4	4	4	4	4	4	4	4
Exp-3	4	4	4	4	4	4	4	4	4	4
Percentage Average	100%	100%	100%	0%	100%	100%	66%	100%	100%	100%
Exp-1 Arabic	4	4	4	4	4	4	3	4	4	4
Exp-2	4	4	4	4	4	4	3	4	4	4
Exp-3	4	4	4	4	4	4	3	4	4	4
Percentage Average	100%	100%	100%	0%	100%	100%	0%	100%	100%	100%

11	12	13	14	15	16	17	18	19	20	Percentage
Po	Li	Po	Po	Ss	Po	Le	Po	Co	Po	
4	4	4	4	4	4	8	4	5	4	80%
4	4	4	4	4	4	8	4	5	4	85%
4	4	4	4	4	4	8	4	5	4	85%
100%	0%	100%	100%	0%	100%	100%	100%	100%	100%	
Average value of Correctness										84%
4	6	4	4	4	4	8	4	5	4	85%
4	6	4	4	4	4	8	4	5	4	85%
4	6	4	4	4	4	8	4	5	4	85%
100%	100%	100%	100%	0%	100%	100%	100%	100%	100%	
Average value of Correctness										85%

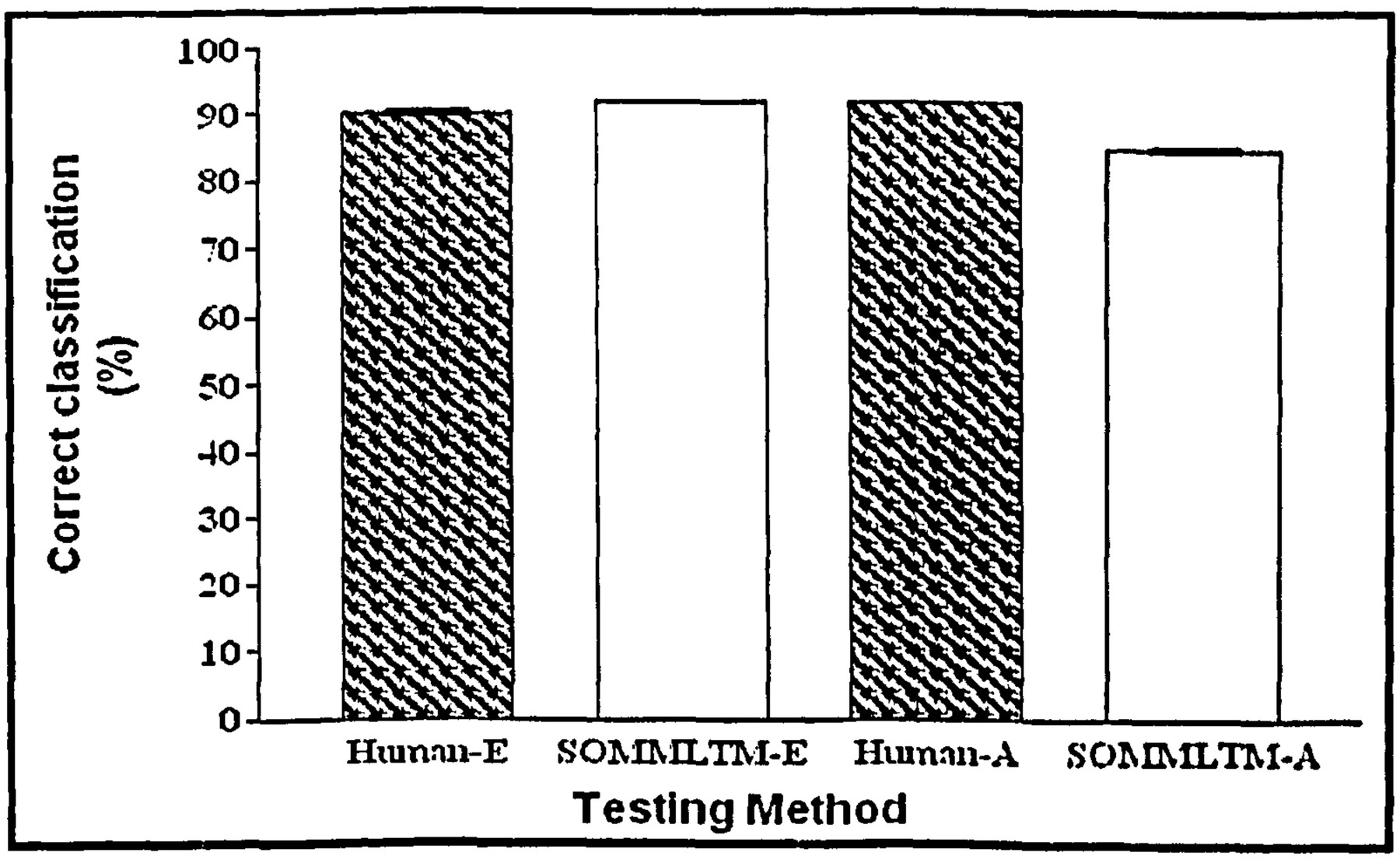


Figure 7.9: Performance of Human Classification vs. MLTextMAES

The Bar chart 7.9 shows the percentage of classification by humans (Hunan-E) in the Arabic language (Human-A) at about 84%, while the precentage for English was 85%. On the other hands, the MLTextMAES (MLTextMAES-A) obtained a precentage in Arabic of 85% correct, and the precentage for English (MLTextMAES-E) was 84%. Table 7.8 shows the mean S.E.M of three observations from three separate experiments. Statistical analysis was carried out using a One-way ANOVA followed by the Bonferroni Multiple Comparison Test to compare all pairs of columns to each other. P values are also shown in Table 7.8.

Table 7.8: Comparison of Human Performance vs. MLTextMAES

Bonferroni's Multiple Comparison Test	Mean Difference	P value
Human-E vs MLTextMAES-E	5%	P < 0.001
Human-A vs MLTextMAES-A	2%	P < 0.001
MLTextMAES-E vs MLTextMAES-A	2%	P < 0.001

The Table 7.8 consists of three tests as shown. The first, related to English documents classified by humans and by the NN model MLTextMAE. The mean difference and the P values are shown as well. Similarly for Arabic documents the corresponding results are depicted in row 2. Next for completeness the developed MLTextMAES model has been tested on Arabic and English documents simultaneously and the relative mean difference and P value are given. As it can be clearly observed that in each case the mean differences and the P values are significantly small, indicating the validity of the developed model. Also, this experiment shows that the model classification is very close to and consistent with the human document classification.

Table 7.9: Comparison of Human Classification vs. MLTextMAES for English Documents

Doc.no English	1	2	3	4	5	6	7	8	9	10
Human	100%	100%	100%	100%	100%	100%	66%	66%	100%	100%
MLTextMAES	100%	100%	100%	0%	100%	100%	66%	100%	100%	100%
Matching	✓	✓	✓	×	✓	✓	✓	×	✓	✓

11	12	13	14	15	16	17	18	19	20	Percentage
0%	0%	100%	100%	66%	100%	100%	100%	100%	100%	
100%	0%	100%	100%	0%	100%	100%	100%	100%	100%	
×	×	✓	✓	×	✓	✓	✓	✓	✓	75%

Table 7.10: Comparison of Human Classification vs. MLTextMAES for Arabic Documents

Doc.no Arabic	1	2	3	4	5	6	7	8	9	10
Human	33%	100%	100%	33%	100%	100%	66%	100%	100%	100%
MLTextMAES	100%	100%	100%	0%	100%	100%	0%	100%	100%	100%
Matching	×	√	√	×	√	√	×	√	√	√

11	12	13	14	15	16	17	18	19	20	Percentage
100%	0%	100%	100%	33%	100%	100%	100%	100%	100%	
100%	100%	100%	100%	0%	100%	100%	100%	100%	100%	
√	×	√	√	×	√	√	√	√	√	75%

Table 7.9 and Table 7.10 show the reliability of MLTextMAES by comparing it to the human performance. The results illustrate the matching of correct results between the human performance and the MLTextMAES model. Both for Arabic and English document clustering 75% match is obtained. This indicates the high level of closeness between the two modes of classification. Figure 7.11 visualizes the results projected into two-dimensional output space, the output is forced to project onto 4 distinct groups, details of which are below.

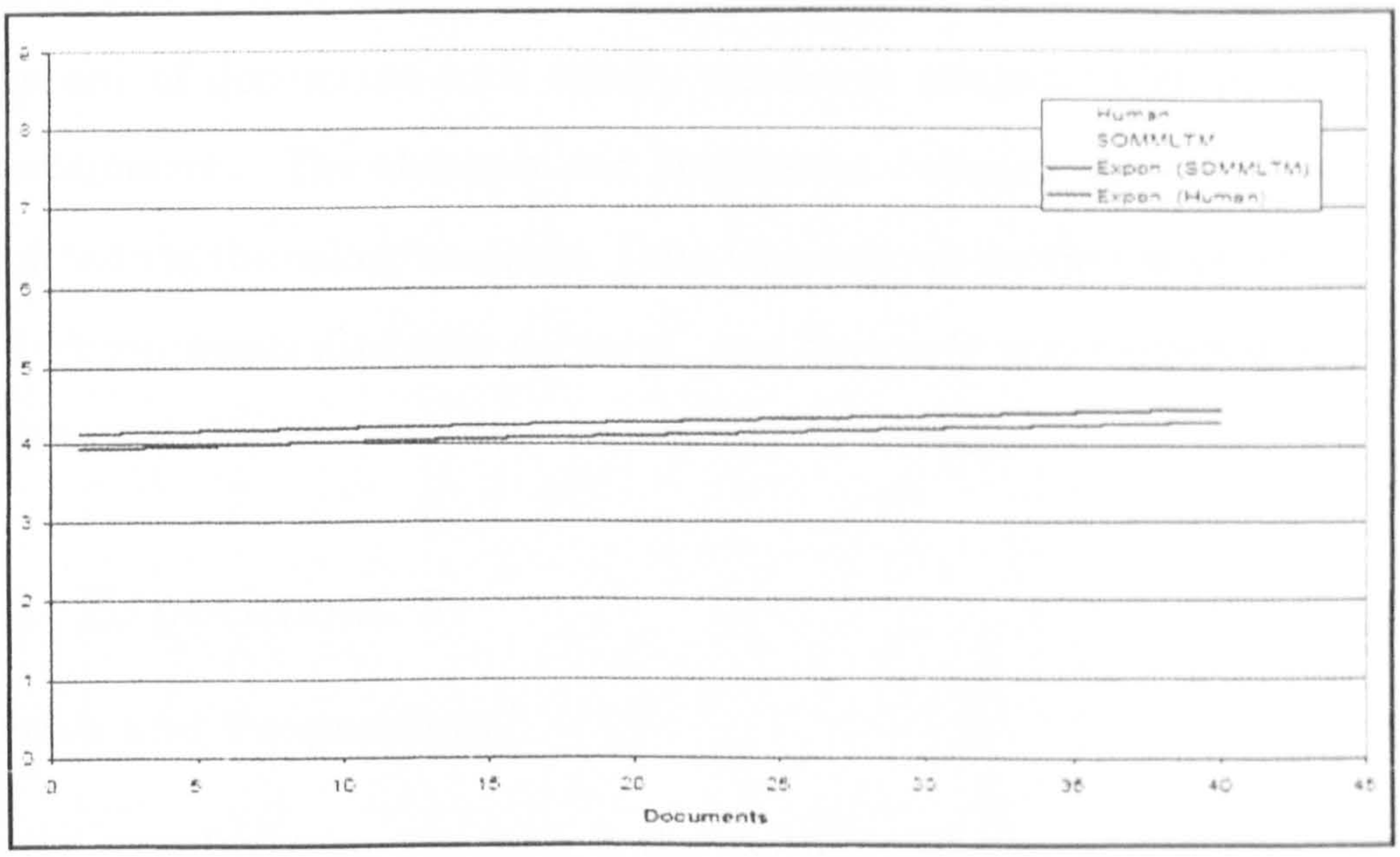


Figure 7.10: The trend of humans and for MLTextMAES model on the validation of classification similarity in the multilingual framework

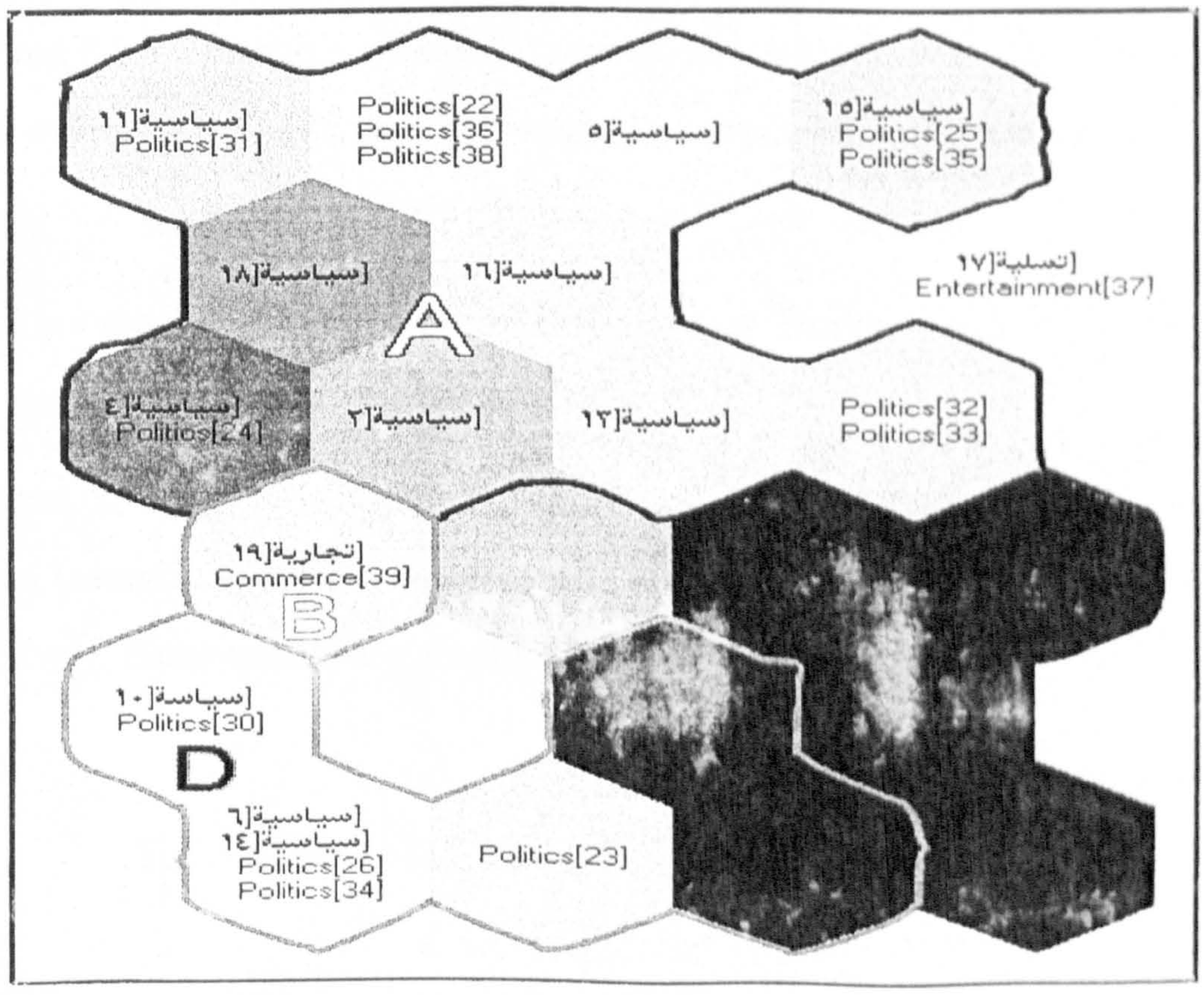


Figure 7.11: Clusters identified on the map

Groups A and D gathers the documents related to politics, while Group B is composed of documents with mainly commerce subject. Group C is related to entertainment. The strength and similarities between the sets of documents are depicted via the colour contours. Pure white shows maximum similarities, while pure Black represents dissimilar subjects. And obviously gray regions show relatively intermediate similarities between the selected documents.

7.4.5 Experiment 5

Objective and Properation

Again the essentially experiment 5 is the extension of the experiment 4 i.e. application of MLTextMAES, on multilingual corpus but with relatively huge datasets. Specifically, it is about corpus composed of 100 Arabic documents. Additionally for each Arabic document of the corpus 7 different English translations (4 from human effect and 3 from machine generated translations) are available. The size of Arabic corpus roughly 15k words, and 14k words per corpus for translations.

Results

Results are compiled for two different categories i.e. human based translation (Table 7.11), and machine based translation (Table 7.12).

Quantization error, percentage of words used in SOM training stage and total cputime (second) spender a valuated and are given in Table 7.11 and Table 7.12, respectively. Same results are graphically depicted in Figure 7.12.

Table 7.11: Average of Quantization Error between Human Translation on LDC Corpus

Scheme	Documents	Quan-Error	Words Percentage	Cputime Sec.
ahd	200	3.52	76%	676.14
ahe	200	3.61	70%	656.49
ahg	200	3.62	69%	677.20
ahi	200	3.59	68%	664.28

Table 7.12: Average of Quantization Error between Machine Translation on LDC Corpus

Scheme	Documents	Quan-Error	Words Percentage	Cputime Sec.
ama	200	3.45	61%	586.29
ame	200	3.52	71%	642.85
arp	200	3.50	68%	634.61

From Table 7.11 and Table 7.12 above, we know that the experiments results show the machine ID ama represents the quantization error 3.45 and cputime 86.29 second compared to human translations. Moreover, our experiments verify the developed model not only depends upon the amount of datasets but it also depends on their translation quality. As it can be clearly observed that in each case the quantization error values are significantly small, indicating the validity our model. Also, this experiment shows that the different translation is very close to each other.

In this case, the results indicate that the machine translation is better than the human translation. The machine translation yielded smaller quantization error in the three different group of translations, and the machine translation obtained better results on cputime (second) compared to the human translation.

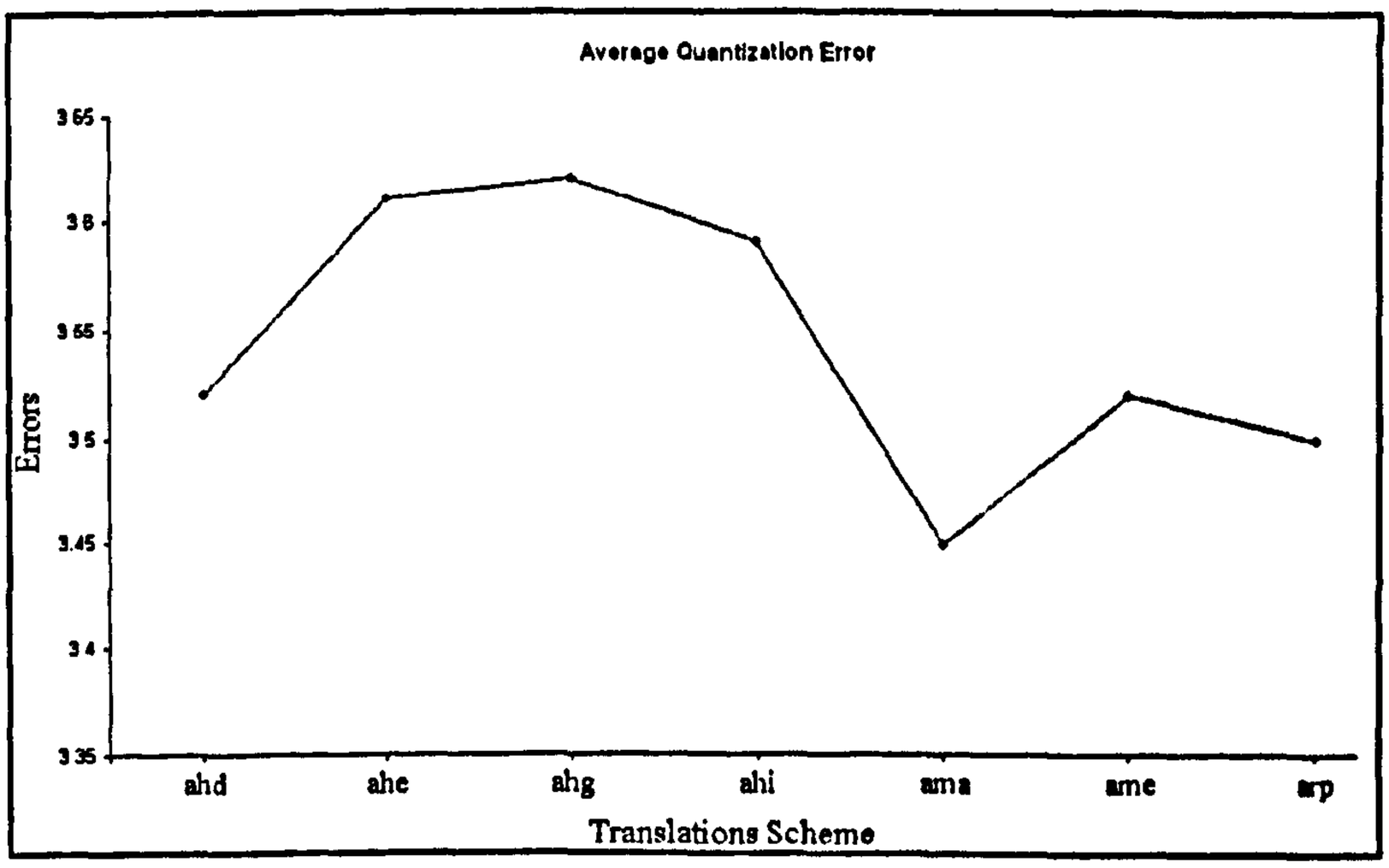


Figure 7.12: Average Quantization Errors For Different Translations

7.5 Evaluative Measures

There are two criteria [160] that could be used to evaluate retrieval system:

1. Recall

2. Precision

These criteria have most frequently been applied in measuring retrieval system, although they have been criticized for a variety of reasons, and a number of alternatives have been suggested. However, recall and precision continue to be used widely despite their shortcomings, partly due to to lack of agreement regarding which measures might be better.

In some studies relevance judgments are allowed to fall into more than two categories, but only a few tests actually take advantage of different relevance levels. More often relevance is conflated into two categories at the analysis phase because of the calculation of precision and recall.

The experiments were performed on various multilingual datasets. The corpus selected were of 20 Arabic documents, 40 Arabic with English translations and then

200 Arabic-English documents. Source of the corpus used in these experiments were from the LDC 2003 news wire in Arabic and its translation in English. Similarity between the documents were computed using the clustering algorithm implemented in the developed model. Testing and evaluation of the retrieval system plays an important role in judging the efficiency and effectiveness of the retrieval process. Several evaluation criteria were used in different experiments, among them are, recall and precision. Recall and precision are measured after the documents are presented on the clustered map. Recall is defined as the ratio of the number of relevant documents that are retrieved to the total number of relevant documents in collection. Precision is defined as the ratio of the number of relevant documents that are retrieved to the total number of documents retrieved in corpus [29].

Table 7.13 gives the Recall and Precision measure for eight corpus of sizes 10, 15, 20, 25, 30, 35 and 40 files. Same results are displayed in graphical format in Figure 7.13.

Table 7.13: Comparative Results for Various Clustering

Documents	Recall	Precision
10	0.5	0.714
15	0.46	0.763
20	0.6	0.774
25	0.6	0.833
30	0.63	0.863
35	0.65	0.884
40	0.73	0.889

As it can be easily observed from Table 7.13 that for relatively large corpus (i.e. size 40) both Recall 0.73 and Precision 0.889 are higher than the small size corpus. This trend if extrapolated, suggests that for even larger corpus Recall and Precision will be even much better.

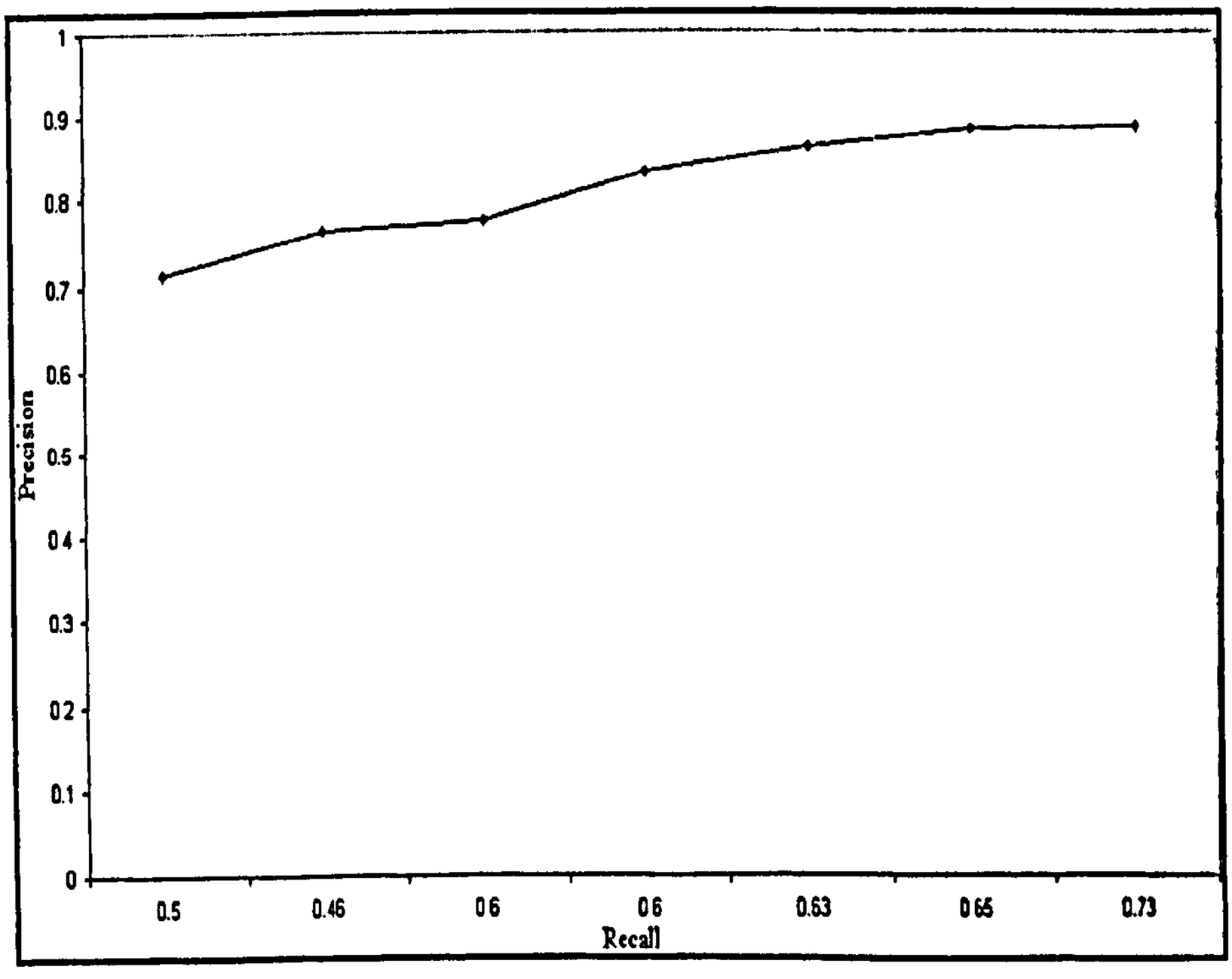


Figure 7.13: Average Recall-Precision

7.6 Existing Tool Support

Currently, in the literature there are a number of tools. The important one being examined here is WEKA tool [161]. This tool will be contrasted with our ML-TextMAES model, which mentioned in Chapter 5.

WEKA is composed of different Machine Learning Algorithms implemented in Java. Main utilization of WEKA is to apply a learning method to a dataset and analyze its output to extract information about the data. Another application is to use several learners and compare their performance in order to choose one for prediction. WEKA is a supervised learning method tool.

It users interface is "command line". Input is accepted in ARFF format only. Thus if a structured data is available (e.g. in MS Excel .csv format) it has to be converted in ARFF layout.

Figure 7.14 shows a user interface developed in JAVA language. In this interface, the user can input his/her desired technique of preprocessing, classify, cluster, associate, select attributes and visualize the results.

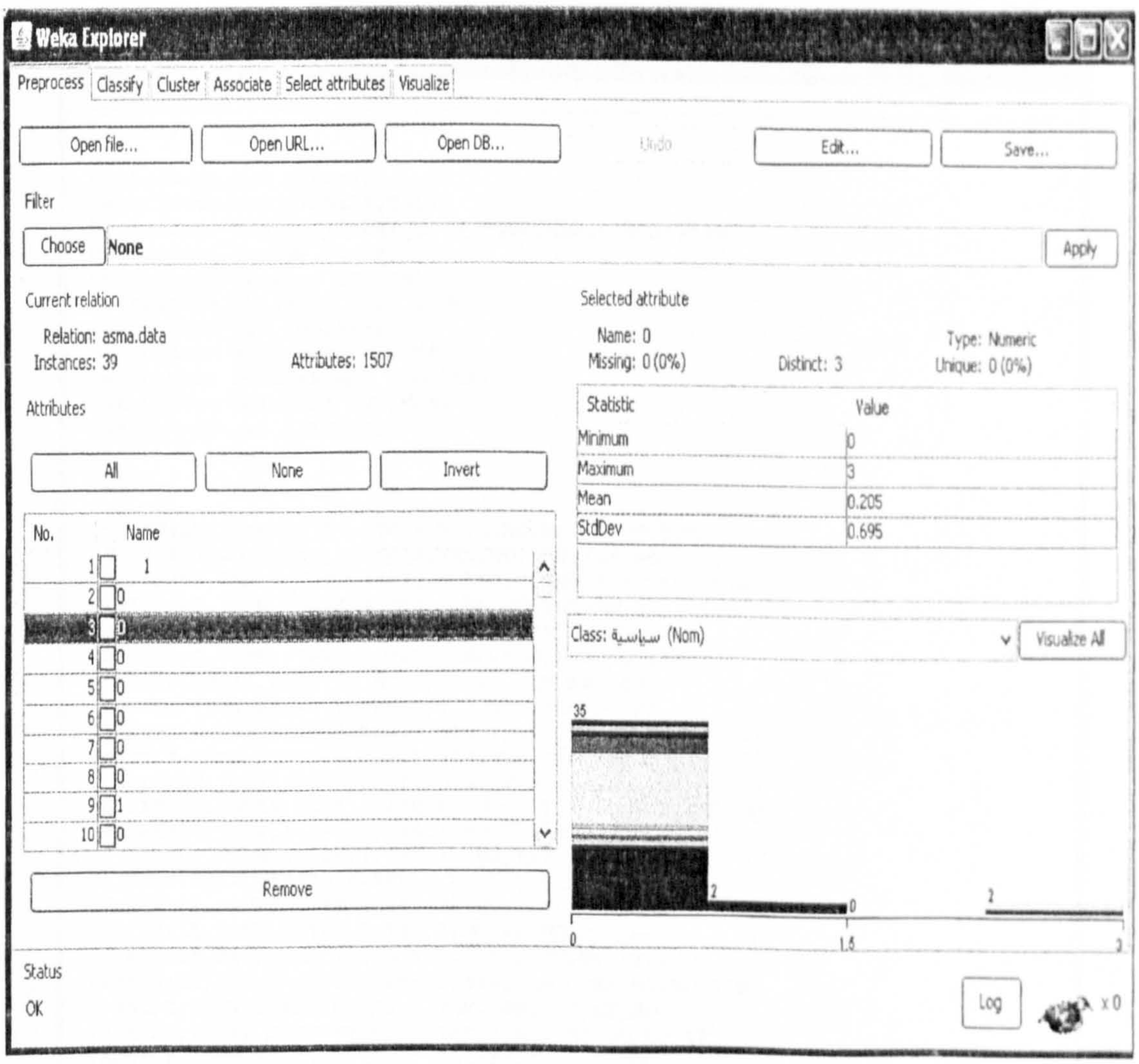


Figure 7.14: User Interface for WEKA Tool

Figure 7.15 shows a sample .arff format file form WEKA examples. It is used to test the WEKA ability to classify and predict the output on the basis of training data. Looking at the file format we notice that attributes are pacified by human to aid in advance the classification process. All together 873 "leaves" (parameters) are measured. The decision leaf is the parameter pep with simple "Black and White" outputs i.e. yes and no. The .csv file is initially converted into ARFF format through

series of steps and then using the command line the generated .arff file is supplied as the input.

bank - WordPad

File Edit View Insert Format Help

```
@relation bank

@attribute age numeric
@attribute sex (FEMALE,MALE(
@attribute region (INNER_CITY,TOWN,RURAL,SUBURBAN(
@attribute income numeric
@attribute married (NO,YES(
@attribute children (0,1,2,3(
@attribute car (NO,YES(
@attribute save_act (NO,YES(
@attribute current_act (NO,YES(
@attribute mortgage (NO,YES(
@attribute pep (YES,NO(

@data

,48FEMALE,INNER_CITY,17546,NO,1,NO,NO,NO,NO,YES
,40MALE,TOWN,30085.1,YES,3,YES,NO,YES,YES,NO
,51FEMALE,INNER_CITY,16575.4,YES,0,YES,YES,YES,NO,NO
,23FEMALE,TOWN,20375.4,YES,3,NO,NO,YES,NO,NO
,57FEMALE,RURAL,50576.3,YES,0,NO,YES,NO,NO,NO
,57FEMALE,TOWN,37869.6,YES,2,NO,YES,YES,NO,YES
,22MALE,RURAL,8877.07,NO,0,NO,NO,YES,NO,YES
,58MALE,TOWN,24946.6,YES,0,YES,YES,YES,NO,NO
,37FEMALE,SUBURBAN,25304.3,YES,2,YES,NO,NO,NO,NO
,54MALE,TOWN,24212.1,YES,2,YES,YES,YES,NO,NO
,66FEMALE,TOWN,59803.9,YES,0,NO,YES,YES,NO,NO
,52FEMALE,INNER_CITY,26658.8,NO,0,YES,YES,YES,YES,NO
,44FEMALE,TOWN,15735.8,YES,1,NO,YES,YES,YES,YES
,66FEMALE,TOWN,55204.7,YES,1,YES,YES,YES,YES,YES
,36MALE,RURAL,19474.6,YES,0,NO,YES,YES,YES,NO
,38FEMALE,INNER_CITY,22342.1,YES,0,YES,YES,YES,YES,NO
,37FEMALE,TOWN,17729.8,YES,2,NO,NO,NO,YES,NO
,46FEMALE,SUBURBAN,41016,YES,0,NO,YES,NO,YES,NO
,62FEMALE,INNER_CITY,26909.2,YES,0,NO,YES,NO,NO,YES
,31MALE,TOWN,22522.8,YES,0,YES,YES,YES,NO,NO
,61MALE,INNER_CITY,57880.7,YES,2,NO,YES,NO,NO,YES
,50MALE,TOWN,16497.3,YES,2,NO,YES,YES,NO,NO
,54MALE,INNER_CITY,38446.6,YES,0,NO,YES,YES,NO,NO
,27FEMALE,TOWN,15538.8,NO,0,YES,YES,YES,YES,NO
,22MALE,INNER_CITY,12640.3,NO,2,YES,YES,YES,NO,NO
```

For Help, press F1 NUM

Figure 7.15: Sample File Bank.arff.

Figure 7.16 is the snapshot of the WEKA output. Cluster is done on the "age" attribute. Randomly selected sub-attribute (sex or income or even no attributes) are also displayed. But the important parameter is the outcome of the leaf "pep" which is necessarily given.

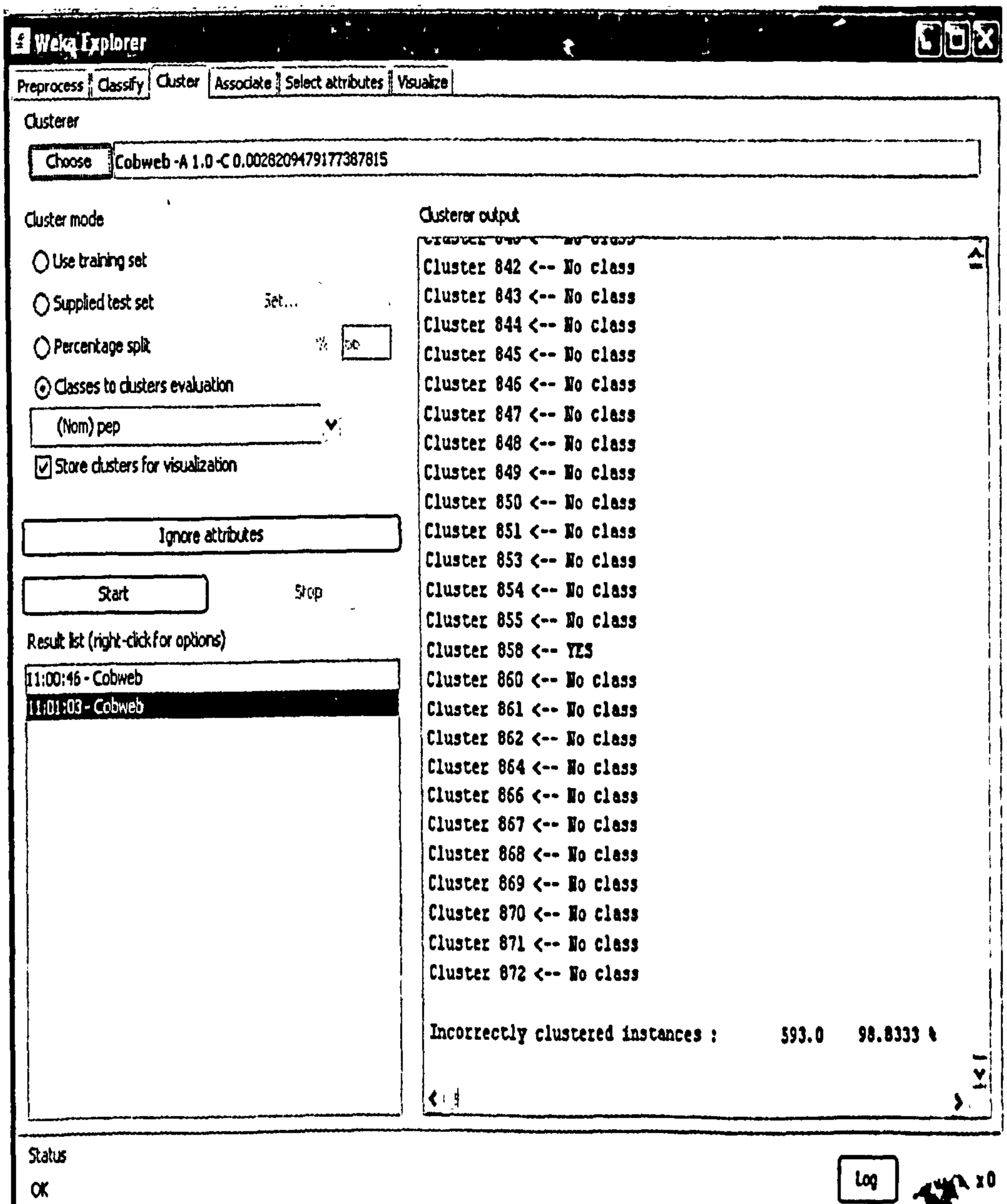


Figure 7.16: Output of WEKA Applied on bank.arff

```
15:47:28 - rules.DecisionTable
              mortgage
              pep
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 600
Number of Rules : 58
Non matches covered by Majority class.
Best first search for feature set,
terminated after 5 non improving subsets.
Evaluation (for feature selection): CV (leave one out)
Feature set: 1,5,6,8,10,11

Time taken to build model: 0.21 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         490               81.6667 %
Incorrectly Classified Instances       110               18.3333 %
Kappa statistic                          0.628
Mean absolute error                      0.2516
Root mean squared error                  0.3804
Relative absolute error                 50.6886 %
Root relative squared error             76.3623 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.755     0.132      0.828     0.755      0.79     YES
  0.868     0.245      0.809     0.868      0.837    NO

=== Confusion Matrix ===

   a   b   <-- classified as
 207  67 |   a = YES
  43 283 |   b = NO
```

Figure 7.17: Output of WEKA Applied on bank.arff

Figure 7.17 is the snapshot of the WEKA output. The scheme of classification is DecisionTable was done on the "age" attribute. On the basis of classification and clustering 600 instances are reported. Number of Rules 58 (DecisionTable). In first phase output of the training section is given which is 81.666 Incorrectly classified instances are given was 18.333. The output of testing phase is unfortunately not very impressive since now in contrast with the very optimistic outcome of the training phase. The clustered instances display the summary of the training and testing phases with 44% (yes) for each instances out of total 600 distributed along the diagonal elements.

7.6.1 Differences between WEKA and MLTextMAES

Now, we can provide analysed comparison between WEKA and MLTextMAES based on our experiments.

1. WEKA supports of data mining, and it is weak in its support for dealing with textual data.

2. WEKA environment requires attributes to be defined in advance for the algorithm to work. MLTextMAES uses free text which has no pre-defined attributes or structure which also required by WEKA.

3. WEKA does not support text categorization very well as reported in Q.F.A. "Most classifiers in WEKA cannot handle String attributes. For these learning schemes one has to process the data with appropriate filters, e.g., the StringToWordVector [2] filter which can perform TF/IDF transformation. The StringToWordVector filter places the class attribute of the generated output data at the beginning".

[2]http://weka.sourceforge.net /doc.dev /weka/ filters/ unsupervised/ attribute/ StringToWordVector.html

4. MLTextMAES tool, on the other hand, support Arabic and English, it is unsupervised learning method and using unstructured dataset, while WEKA do not support Arabic language.

5. WEKA algorithm is useful for structured data with data present in single file along with the related attributes. While MLTextMAES deals with multi-files which is unstructured.

6. WEKA tool accepts a single structured input file for its analysis. No clustering between the documents is generated by WEKA. Thus no comparison between MLTextMAES and WEKA in terms of proper clustering or accuracy or efficiency is possible.

The WEKA tool is tested on a single file which we have provided in .ARFF file format. The experiments were conducted on 32 instances are depicted in Figure 7.18. The learning and testing data were obtained in the manner described in Section 7.6. Figure 7.19 shows the results of the classification obtained from this experiment was 65.6 correct, and incorrect classification was 34.5. Also, the confusion matrix displays the summary of the training phase with 11 Arabic instances and 21 English instances out of total 32 distributed along the diagonal elements. Error analysis shows the recall, precision and f-measure statistical parameters are zero in Arabic. The output of predictive phase is unfortunately not very impressive since now in contrast with the very optimistic outcome of the training phase. To make a sensible comparison, we imitate what happen in our system by providing the following data.

data-doc - WordPad

File Edit View Insert Format Help

```
@relation document-cluster

@attribute files {document{
@attribute id real
@attribute class {appl_science,nat_science,socia_science,arts,life,leisure,commerce,politics}
@attribute Lnaguage {Arabic, English{

@data
document,1,appl_science,Arabic
document,2,politics,English
document,3,life,Arabic
document,4,commerce,Arabic
document,5,politics,English
document,6,socia_science,English
document,7,commerce,English
document,8,nat_science,English
document,9,arts,Arabic
document,10,appl_science,English
document,11,arts,Arabic
document,12,life,English
document,13,leisure,Arabic
document,14,nat_science,English
document,15,commerce,English
document,16,appl_science,Arabic
document,17,leisure,English
document,18,nat_science,English
document,19,commerce,English
document,20,socia_science,English
document,21,politics,English
document,22,life,English
document,23,socia_science,English
document,24,arts,Arabic
document,25,appl_science,Arabic
document,26,commerce,Arabic
document,27,nat_science,English
document,28,politics,Arabic
document,29,leisure,English
document,30,socia_science,English
document,31,politics,English
document,32,nat_science,English
```

For Help, press F1 NUM

Figure 7.18: Sample File docs-data.arff

```
21:44:28 - rules.DecisionTable
Attributes:    4
               files
               id
               class
               Lnaguage
Test mode:     evaluate on training data

=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 32
Number of Rules : 1
Non matches covered by Majority class.
Best first search for feature set,
terminated after 5 non improving subsets.
Evaluation (for feature selection): CV (leave one out)
Feature set: 4

Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          21               65.625  %
Incorrectly Classified Instances        11               34.375  %
Kappa statistic                          0
Mean absolute error                      0.4512
Root mean squared error                  0.475
Relative absolute error                 99.3674 %
Root relative squared error             99.9813 %
Total Number of Instances               32

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0         0          0         0         0        Arabic
  1         1          0.656     1         0.792    English

=== Confusion Matrix ===

  a  b   <-- classified as
  0 11 |  a = Arabic
  0 21 |  b = English
```

Figure 7.19: Output of WEKA Applied on docs-data.arff

7.7 Efficiency of the Model

The performance tests were made on a computer with Pentium(R), 512MBs of memory and a 1.86 GHz CPU, running a Windows XP operating system. Additionally, some tests were performed with 1.0 GHz CPU, Pentium III and 256MBs of memory, but also with a Windows XP operating system. The Matlab version in the environment was 7.1.0 (R14).

The purpose of the performance tests was only to evaluate the computational load of the algorithms. No attempt was made to compare the quality of the resulting mappings, primarily because there is no uniformly recognized "correct" method to evaluate them with. The tests were performed with data sets and maps of different sizes, and two different training functions were tried on this problem: batch-train and seq-train (see Section 5.5). The SOMMLTM algorithm was tested on Arabic-English documents. The training performances of the SOMMLTM algorithm, together with times in CPU seconds, are listed in Table 7.14 below.

Table 7.14: The Timing Performance of SOMMLTM

Machines	Data Size	Docs no	Batch-train (Sec)	Seq-train (Sec)	Efficiency
Intel Pentium(R)	7300	20	14.17	48.93	3.45%
	27040	40	20.2	88.27	4.37%
	52740	60	25.6	132.02	5.15%
	78480	80	30.05	156.2	5.19%
Pentium III	7300	20	67.07	366.04	5.45%
	27040	40	103.58	817.16	7.89%
	52740	60	138.10	1140	8.25%
	78480	80	156.64	1440	9.19%

Table 7.14 shows some typical computing times. Here, the SOMMLTM training time are scaled by using a cputime function. As a general result, in Pentium(R), batch-train was roughly a factor of between three and five times faster than seq-train, increasing with larger input datasets. In Pentium III, batch-train is again much faster than seq-train whose slower speeds clearly came into play. Thus the differences between the training functions were large. In this case the result of the training performance shows that the batch-train is better than seq-train function. Additionally, the batch-train function also handled the large datasets more efficiently than seq-train function as evedint of last column of Table 7.14.

7.8 Summary

Experiments were successfully performed using our formulated model. The ASMA matrix, generated through the first stage of our framework, has been used as an input to the SOM algorithm. We have successfully projected a very high dimensional input (the length of each vector was around 1500 units) onto a two-dimensional output space using standard SOM techniques. In the evaluation stage we have compared the results from our generated code MLTextMAES with the human decisions on both raw and annotated corpora. High level of match between the human and the model based clustering is observed. The results are depicted in Table 7.9 and Table 7.10.

On average, we can see that the results obtained from MLTextMAES model follow the same trends as those obtained from human decisions. It is interesting to notice how similar the human and machine decisions are. This test shows that the model classification is very close to and consistent with human document classifications. In our experiment we distributed the documents only to highly educated people due to our limitations on the class of people available (see Figure 7.10). Overall, our experiment verifies the fact that our developed model predicts quality clustering as achieved by humans. To establish our findings more profoundly, we propose that in some future work a large number of human subjects from diverse backgrounds be utilized and their classifications compared with a relatively large multilingual corpus, using MLTextMAES. Figure 6.11 shows the main results of this study. Here 40 documents (20 Arabic and 20 English) are successfully categorized into 8 main domains. As elaborated previously, we have obtained the average quantization error of 6.834. Statistical analysis have also been done to evaluate our model and results. Accordingly we are 95% confident that correct results will lie within (μ= 2.01915 $\pm$ σ = 1.49828). Table 7.6 and Table 7.7 show the results of this study. The level of significant difference was set at $P < 0.05$. This shows that results are fairly close

from MLTextMAE and the human classification, both for Arabic and English sets of documents.

The performance of the developed SOMMLTM model is evaluated on the basis of using either seq-train or batch-train functions. It is observed that while batch-train is much faster in executing (small CPU time) but little improvement in generated clusters is seen with changing initial parameters (e.g. radius, learning rate, etc.). While seq-train is more CPU consuming but large improvement over results is observed with adjusting the initial parameters.

The general aim was to develop text mining tools to cover both Arabic and English simultaneously, and to implement them through the MLTextMAES technique. The results show that this aim has been achieved successfully (see Figure 7.11 above).

Chapter 8

Conclusions and Future Research

This thesis has focused on issues in the design and implementation of the ML-TextMAES classification system i.e. clustering of Arabic and English documents. We have demonstrated how SOM is capable of clustering textual data that has been converted to numeric data making it particularly useful to researchers in the field of text mining.

We propose that the Arabic stemmer algorithm can be applied to other languages such as Urdu and Persian. This is due to the similarities between the Arabic, Urdu and Persian languages, which all use the same alphabet and calligraphy as that of Arabic. The main aim of developing a multilingual morphology processor is to support the root method of search. The root method is a novel approach and is introduced in this study as it is believed to improve text mining. Furthermore, the root method retrieves almost all relevant texts.

In assessing the performance of MLTextMAES, comparisons with human experts were very important for this work. SOM technique applied on monolingual text mining is common nowadays for many of the world's languages. SOM comparisons with human expertise are commonly performed in order to find areas of improvement.

The result shows that the percentage of classification by humans (orange bar) in the Arabic language was about 84%, while the precentage for English was 85%. On the other hands, the MLTextMAES (yallow bar) obtained the precentage in Arabic of 84% correct, and the precentage for English was 85%. The results shown are the mean S.E.M of three observations from three separate experiments. Statistical analysis was carried out using a One-way ANOVA followed by the Bonferroni Multiple Comparison Test to compare all pairs of columns to each other.

Today, the data required for multilingual text mining is easily available on Internet. Most universities, companies and news agencies (resources) publish reports, or their information, on Internet. The Internet has indeed become a main source of textual data. However, with the almost infinite access to information comes a problem: our capacity to process the information is not sufficient. Indeed, we are need of tools that can help us to mining these data. One such tool is text data mining. text data mining is a tool for extracting non-trivial patterns from unstructured text in large amounts of data. SOM is an example of a text data mining tool considered to be suitable for exploratory data and analysis, particularly clustering and visualization problems. SOM is two-dimensional output of data, which group the data according short distances, or similarities in the dataset. The result is a two-dimensional topological grid map with light shades which indicate the small distances, and dark shades are indicated long distances, or differences in dataset.

A literature review was conducted to study the applications in which SOM has been used. The current status of Natural Language Processing was also investigated in order to determine the methods currently being used. Many researchers are no doubt satisfied with the available tools, particularly in the field of monolingual text mining, but there is a pressing need for alternative methods for the analysis of multilingual text mining.

In this thesis, a new model for the multilingual text mining of data sources has

been proposed together with a new Arabic-English morphological analysis system. In order to evaluate the correctness of our MLTextMAES model we developed an evaluation methodology. This methodology is based on a comparison between the model's output and human outputs.

The current implementation of the MLTextMAES can be enhanced in several ways by changes in the implementing programming language. The $SOM - DOCS - ASMA$ function enables this mission flexibility with the current Matlab implementation. However, our methodology can be realized very efficiently and without the need of complex error-handling routines as required by the sequential cyclical executive approach. The biggest advantage of the SOM algorithm in text data mining applications is its efficiency in clustering large data sets.

Thus, using the constructive research approach, our MLTextMAES model was developed and evaluated. Firstly, the key concepts in multilingual analysis (Arabic-English), knowledge discovery and extraction from datasets, text data mining, and SOM is presented. Then, multilingual datasets for Arabic and English were collected from LDC from January 2003, and from ICA, using the Internet as the source of information.

We conducted a test on the gathered corpus in order to check its validity. Then, a text data mining tool, SOM, was used to perform clustering of these documents. Finally, we developed our model and evaluated it through experts in the subject matter in a face validation setting.

A hexagonal 6X4 map was created for 16 documents, and updated with data from 40 documents. The map was visualized through bilingual documents. The conclusion was that SOMMLTM performance was considerably better at clustering and visualization than other methods. Human classification displayed a stronger performance during the experiment when employing highly educated people, but the performance could be affected by lesser educated people, as well as by larger amounts of docu-

ments.

Experiments were successfully performed using our newly formulated model. The ASMA matrix, generated through the first stage of our framework, was used as an input to the SOM algorithm. We have successfully projected a very high dimensional input (the length of each vector was around 1500 units) onto a two-dimensional output space using standard SOM techniques. On average, we can see that the results obtained from MLTextMAES model follow the same trends as those obtained from human decisions. It is interesting to notice how similar the human and machine decisions are. This test shows that the model classification is very close to and consistent with human document classifications. Overall, our experiment verifies the fact that our developed model predicts the same quality clustering as achieved by humans. We obtained a quantization error of 6.834. Statistical analysis were also conducted to evaluate our model and results. Accordingly we are 95% confident that correct results will lie within (μ= 2.01915) and (σ = 1.49828) respectively.

Table 7.6 and 7.7 show the results of this study. The level for a significant difference was set $P < 0.05$. This shows that the results from MLTextMAES and the human classifications are fairly close, both for the Arabic and English sets of documents.

Several experiments were conducted. The results observed in the first experiment were satisfactory; it showed that 80% of the translations were correct. Then, the model output was improved by solving the problems encountered in this experiment. In the second experiment the model was run again on the same test set; this resulted in identical classifications as in the third experiment. In the third experiment the model was tested on same documents, showing that 85% of the classifications were correct. Overall, the model was validated by the multilingual testing methodology.

The general aim was to develop text mining tools to cover both Arabic and English simultaneously, and to implement them through the SOMMLTM algorithm. The results show that this aim has been achieved successfully.

The performance of the developed SOMMLTM model is evaluated on the basis of using either seq-train or batch-train functions. It is observed that while batch-train is much faster in executing (small CPU time) but little improvement in generated clusters is seen with changing initial parameters (e.g. radius, learning rate, etc.). While seq-train is more CPU consuming but large improvement over results is observed with adjusting the initial parameters as evedint of last column of Table 7.14.

In addition to the presentation of this thesis, material from the research has been presented at conferences and in journal papers. The first paper was presented as a chapter in [125]. This introduces the new matrix $AMESD$, and shows how it is constructed by using Arabic and English morphology and a bilingual dictionary. This paper also details the advantageous features of SOM and backpropagation. The next conference paper [162] presents the results of training the basic model of MLTextMAES to test the performance of calculating the similarities between vectors and neighbourhoods. The model is evaluated using correlation co-efficients, and the similarities are estimated through the cosine of the angle between the vectors.

The main aim of this work, that the SOM is an efficacious tool for multilingual text mining. The primary advantage of the SOM in this case was, as expected, its cluster analysis and visualization properties.

8.1 Limitations of the Study

From the perspective of this study, there are a few, specific limitations in this research which should be addressed as a means for improvement or potential strategies for further study.

Firstly, it is evident that we chose SOM as the main tool in this study for a number of positive reasons, stated earlier. However, SOM has its own intrinsic limitations such as a relatively large quantization error [163], its static nature [164], etc. Apart

from these, the large computational time forced us to make certain choices when using the SOM network e.g.

- Selection of neighbourhood radius R.

  Obviously with a relatively large R, better learning can be expected but this is at the expense of computational time. We therefore selected $R = 3$ and eventually dropped it with iterations to $R = 0$.

- Models for the learning rate.

  Certainly various models for monotonically decreasing are available but it is not possible to explore all of them and evaluate their influence on SOM clustering. Having said this we still looked into three different models for this research and examined the outputs.

- Size of Arabic-English corpora.

  Computational time restricted us to using a relatively small sized corpus for each test study. This we understand could affect the evaluation of our framework, but qualitatively we believe the results are fairly good.

- Arabic-English Dictionary.

  In our indexing and clustering process we used an Arabic-English dictionary that is composed of roots of verbs only, and we did not consider nouns in this study. The obvious reason for this decision is that the majority of nouns cannot be decomposed into roots, which is the basic criterion in our study. The effect of dropping nouns could in our perception cause slight changes in the clustering of Arabic-English documents. This would be especially true for those sets of documents in which large numbers of nouns are repeated.

8.2 Future Research

We have several on-going activities, all concerned with extending our thesis work to be more powerful and applicable. This situation is becoming ever more urgent with the increasing amounts of information available through the Internet. We simply cannot cope with this information overload any longer without using intelligent tools. We present some possible activities as follows:

- Implement the semantic analysis phase, which evaluates the meaning of an individual text. This task is difficult because of the concept of acronyms in natural languages. This phase is essential for solving semantic ambiguities.

- In this study, we have implemented our MLTextMAES model in three stages. Future work might be extended to model and implement the graphical user interface (GUI) as a fourth stage.

- In this study, we have used the human based classification to evaluate the develop model (MLTextMAES) . Future work might be extended to the evaluation through alternative technique e.g. k-means.

- Future work could also include more efficient algorithm for the entire process in text mining.

- Future work could also be extended to a relatively more comprehensive "root dictionary".

- Future work could also be extended to root dictionaries for Urdu and Persian languages.

References

[1] T. Honkela, "Self-Organizing Maps in Natural Language Processing," *ESPOO-Thesis*, 1997.

[2] Kohonen, T., "Self-Organizating Maps," *New York : Springer-Verlag*, p. 86, 1997.

[3] M. A. Hearst, "Untangling Text data mining," *Presentation of ACL'99: the 37th Annual Meeting of Association for Computational Linguistics*, pp. 3-10, 1999.

[4] Han, J. and Kamber, M., "Data Mining : Concepts and Techniques," *Morgan Kaufmann*, 2000.

[5] J. Vesanto, J. Himberg, E. Al-honiemi, and J. Parhankangas, "SOM Toolbox 2.0," *Helsinki University of Technology Helsinki. Finland*, 2000. http://www.cis.hut.fi/projects/somtoolbox/ (Accessed: 02/08/2007).

[6] F. Katamba, "Morphology," *London, Macmillan Press Ltd*, 1993.

[7] S. Al-Saleh, "دراسات في فقه اللغة," *Dar Al-Elm Lilmlaeen-Labanon*, 1989.

[8] I. Al-kharashi, "A microcomputer-based Arabic Information retrieval system comparing words, stems, and roots as index terms," *Ph.D. Thesis. Chicago: Illinios Institute of Technology*, 1991.

[9] H. Abu Salem, "A microcomputer-based Arabic Bibliographic Information retrieval system with relation thesaurus (Arabic-IRS)," *Ph.D. Thesis. Chicago: Illinios Institute of Technology*, 1992.

[10] I. Hmeidi, I., "Design and Implimentation of Automatic word and phrase indexing for Information retrieval with Arabic documents," *Ph.D. Thesis. Chicago: Illinios Institute of Technology*, 1995.

[11] X. Software, "JCreator is a powerful IDE for Java," ., 2007. http://www.jcreator.com/download.htm(Accessed: 20/5/2007).

[12] Tan, Ah-Hwee., "Text Mining: The state of the art and the challenges," *In proceedings, PAKDD'99 Workshop on Knowledge discovery from Advanced Databases (KDAD'99), Beijing,* pp. 71–76, 1999.

[13] D. W. Oard, "The TREC-2002 Arabic/English CLIR Track. ," *The Eleventh Text Retrieval Conference, TREC 2002; Gaithersburg, MD; USA;*, vol. 15, pp. 17–26, 2002.

[14] Feldman, R. and Dagan, I., "Knowledge discovery in textual databases (KDT)," *Canada, August 20-21, AAAI Press*, pp. 112–117, 1995.

[15] W. Fan, L. Wallace, S. Rich, and Z. Zhang, "Tapping the Power of Text Mining," *Communications of the ACM*, vol. 49, no. 9, pp. 76–82, 2006.

[16] J. Maloney, "Text Mining Solutions," *SRA International, Inc.*, 2005.

[17] C. Lee and H. Yang, "A Multilingual Text Mining Approach Based on SOM," *Kluwer Academic Publishers, Netherlands*, vol. 18, pp. 295–310, 2003.

[18] M. Mladenic, D.and Grobelnik, "Overview of Text Mining and Web Mining," *Proceedings of the 4th International Multi-conference Information Society*, vol. A, pp. 46–85, 2001.

[19] S. Dzeroski, L. D. Raedt, and S. Wrobel, "Multirelational Data Mining 2003: Workshop Report," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 2, pp. 200–202, 2003.

[20] F. Thabtah, P. Cowling, and Y. Peng, "Comparison of Classification Techniques for A Personnel Scheduling Problem," *Proceedings of IBMA International Conference, Amman, Jordan.*, 2004.

[21] F. Neri, "Text Mining Solution: A New Way to Explore Patent Database," *Lexical Systems Lab-Synthema, Italy*, 2004.

[22] D. Wohl, A., "Intelligent Text Mining Creates Business Intelligence," *White paper, IBM*, 1998.

[23] w. j. Tukey, "Exploratory Data Analysis," *Addison-Wesley Publishing Company*, 1977.

[24] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, "Understanding Robust and Exploratory Data Analysis," *John Wiley & Sons, Inc.*, 1983.

[25] T. Eldos, "Arabic Text Data Mining: A Root-Based Hierarchical Indexing Model. ," *International Journal of Modelling and Simulation,,* vol. 23, no. 3,, pp. 158–166, 2003.

[26] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Advances in Knowledge Discovery and Data Mining," *AAAI/MIT Press*, 1995.

[27] M. Rajman and Besancon, "Text Mining: Natural Language Techniques and Text Mining Application.," *IFIP. Chapman & Hall* ,, vol. 23, no. 3, pp. 158–166, 1998.

[28] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys,,* vol. 34, no. 1, pp. 1–47, 2002.

[29] G. Salton and J. McGill, M., "Introduction to Modern Information Retrieval," *McGraw-Hill*, pp. p62–63, (1983).

[30] M. Aman, " Use of Arabic in Computerised Information Interchange," *Journal of the American Society of Information Science*, vol. 35(4), pp. 204–210, 1984.

[31] d. Baek, H. Lim, and H. Rim, "Tapping the power of text mining," *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.

[32] M. Kobayashi and K. Takeda, "Information Retrieval on the Web," *ACM Computing Surveys (CSUR),*, vol. 32, no. 2, 2000.

[33] K. S. Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments," *Information Processing and Management: an International Journal*, vol. 36, no. 6, 2000.

[34] G. Amatia, "Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness," *ACM Transactions on Information Systems,*, vol. 20, no. 4, pp. 357–389, 2002.

[35] H. Abu-Salem, M. Al-Omari, and W. Evens, M., "stemming Methodologies Over Individual Query Words for an Arabic Information Retrieval System," *Journal of American Society for Information Science*, vol. 50(60), pp. 524–529, 1999.

[36] K. Papineni, "Why inverse document frequency? ," *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pp. 1–8, 2001.

[37] Y. Wilks, "Information extraction as a core language technology," *Pazienza, editor, Information Extraction. Springer, Berlin*, 1997.

[38] S. L. Larkey, L. Ballesteros, and E. M. Connell, "Light Stemming for Arabic Information Retrieval,"

[39] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrachs, and R. Shamir, "An Algorithm for Clustering cDNAs for Gene Expression Analysis," *In RECOMB*, vol. 17, pp. 188–197, 1999.

[40] D. G. Wastell and R. Gray, "The Numerical Approach to Classification: A Medical Application to Develop a Typology for Facial Pain," *In Statistics in Medicine*, vol. 6, pp. 137–164, 1987.

[41] I. Jolliffe, B. Jones, and B. J. T. 0Morgan, "Utilising clusters: A Case Study Involving the Elderly," *In Journal of Roy. Statist. Soc.*, vol. 145, pp. 224–236, 1982.

[42] F. Daniel, "An Analysis of Recent Work on Clustering Algorithms," *echnical report, University of Washington*, 1999.

[43] M. Berry and G. Linoff, "Data Mining Techniques for Marketing, Sales, and Customer Support," *John Wiley and Sons, New York*, 1997.

[44] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques," *Journal of Intelligent Information Systems*, vol. 17:2/3, pp. 107–145, 2001.

[45] M. Chen, J. Han, and P. S. Yu, "Data mining: An Overview from a Database Perspective.," *IEEE Transactions On Knowledge And Data Engineering,*, vol. 8, no. 6, pp. 866–883, 1996.

[46] A. A. Freitas, "A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery," *Advances in evolutionary computing: theory and applications.*, pp. 819–845, 2003.

[47] J. G. Deboeck, "Financial Applications of Self-Organizing Maps," *American Heuristics Electronic Newsletter*, pp. 1–7, 1998.

[48] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, , and A. Saarela, "Self organization of a massive document collection," *IEEE Transactions of Neural Networks*, no. 11(3), pp. 574–585, 2000.

[49] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "WEBSOM- Self-organizing maps of document collections," *Neurocompting*, vol. 21(1-3), pp. 101–117, (1998).

[50] T. Kohonen, "Self-Organizing Maps," *Springer*, (1995).

[51] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *In Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, Univ. of California Press*, vol. 1, pp. 281–297, 1967.

[52] P. N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," *Addison-Wesley*, vol. Ch-8, 2005.

[53] Y. Xinhua, Y. Kuan, and D. Wu, "A k-means Clustering Algorithm based on Self-Adoptively Selecting Density Radius ," *IJCSNS International Journal of Computer Science and Network Security,*, vol. 6, no. 8A, pp. 43–46, 2006.

[54] K. T. Moon, "The Expectation Maximization Algorithm," *Signal Processing Magazine, IEEE*, vol. 13, no. 6, pp. 47–60, 1998.

[55] B. Bailey and C. Elkan, "Fitting A Mixture Model by EXPECTATION MAXIMIZATION to Discover Motifs in Biopolymers," *UCSD Technical Report CS94-351*, pp. 10–32, 1994.

[56] J. Salojarvi, K. Puolamaki, and S. Kaski, "Expectation Maximization Algorithms for Conditional Likelihoods," *Proceedings of the 22 nd International Conference on Machine Learning, Bonn, Germany,,* 2005.

[57] M. Baglioni, B. Furletti, and F. Turini, "DrC4.5: Improving C4.5 by Means of Prior Knowledge," *Proceedings of ACM symposium on Applied computing, Santa Fe, New Mexico.*, pp. 474–481, 2005.

[58] S. Nijssen and E. Fromont, "Mining Optimal Decision Trees from Itemset Lattices," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, San Jose, California, USA*, pp. 530 – 539, 2005.

[59] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "DistributionalWord Clusters vs. Words for Text Categorization," *The Journal of Machine Learning Research, MIT Press Cambridge, MA, USA*, vol. 3, pp. 1183 – 1208, 2003.

[60] W. W. Cohen, "Fast Effective Rule Induction," *Proceeding of the 12th International Conference, ML95*, 1995.

[61] W. W. Cohen, "Learning States and Rules for Time Series Anomaly Detection," *American Association for Artificial Intelligence*, 2004.

[62] E. Zohar, Y., "Introduction of Text Mining," *Supercomputing*, 2002.

[63] C. Metz, "Software: Text Mining," *in PC Magazine*, 2003.

[64] N. Kostoff, R., "BioMedical Literature Text Mining," *Office of Naval Research, 800 N. Quincy St., Arlington, VA 22217, Washington DC*, 2004.

[65] N. Treloar, "Text Mining: Tools, Techniques, and Applications," *AvaQuest*, 2002.

[66] E. Gaussier, "Processing Multilingual Collections for text Mining Applications," *Xerox Research Centre Europe*, vol. The world's knowledge www.bl.uk, 2004.

[67] W. M. Berry, "Survey of Text Mining: Clustering, Classification, and Retrieval," *Springer-Verlag , New York*, pp. 133–137, 2004.

[68] Kohonen, T., "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43:, pp. 59–69, 1982.

[69] R. Besancon and M. Rajman, "Evaluation of a vector space similarity measure in a multilingual framework," *Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland*, 2004.

[70] Azzam, S. and Humphreys, K. and Gaizauskas, R. and Wilks, Y., "Using a Language Independent Domain Model for Multilingual Information Extraction," *Applied Artificial Intelligence*, vol. 13, pp. 705–724, 1999.

[71] R. Chau and C. Yeh, "A multilingual Text Mining Approach to Web Cross-lingual Text Retrieval," *Knowledge based systems*, vol. 17, pp. 219–227, 2004.

[72] Q. Ma, K. Kanzakib, Y. Zhangb, M. Muratab, and H. Isahara, "Self-organizing semantic maps and its application to word alignment in JapanseChinese parallel corpora," *Neural Networks*, vol. 17, p. 12411253, 2004.

[73] F. Neri and R. Raffaelli, "Five Steps to Text Mining in Biomedical Literature," *Springer Berlin / Heidelberg*, vol. 185/2005, pp. 123–131, 2006.

[74] T. T. and C. Zhai, "Mining Comparable Bilingual Text Corpora for Cross-Language Information Integration," *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. Chicago, Illinois, USA*, pp. 691–696, 2005.

[75] L. Denoyer, S. Brunessaux, J. Vittaut, P. Gallinari, E. S&DE, and S. Brunessaux, "Structured Multimedia Document Classification," *Proceedings of the ACM symposium on Document engineering Grenoble, France*, pp. 153–160, 2003.

[76] S. Montalvo and A. F. V. Martne, R.and Casillas, "Multilingual Document Clustering: an Heuristic Approach Based on Cognate Named Entities," *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL. Sydney, Australia*, pp. 1145–1152, 2006.

[77] B. Mathiak and S. Eckstein, "Five Steps to Text Mining in Biomedical Literature," *In proc. of the 2 European Workshop on Data Mining and Text Mining for Bioinformatics, Pisa. Italy.*, pp. 47–50, 2004.

[78] E. Norman and C. Sondak, "Neural networks and artificial intelligence," *ACM 0-89791-298-5/89/0002/0241*, 1989.

[79] A. L. Wilkes and N. J. Wade, "Bain on neural networks," *Brain and Cognition*, vol. 33, no. :, pp. 295–305, 1997.

[80] B. Alexander, "The columbia encyclopedia," *Prentice Hall International, Inc., Upper Saddle River, N.J.*, no. Sixth Edition, 2006.

[81] W. S. McCulloch and W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5:, pp. 115–137, 1943.

[82] D. Olmsted, "History and principles of neural networks to 1960," *Net Objects Vusion 7*, 1998. http://www.neurocomputing.org/html/nn_1960.html (Accessed: 02/06/2006).

[83] Fausett L., "Fundamentals of Neural Networks," *Prentic Hall, Inc.*, 1994.

[84] T. Kohonen, "Correlation matrix memories," *IEEE Transactions on Computers C-21*, no. 353-359, 1972.

[85] H. Ritter, T. Martinetz, and K. Schulten, "Neural computation and self-organizing maps," *Addison Wesley Publishing Company*, 1992.

[86] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[87] M. Diab, "An Unsupervised Method for Multilingual Word Sense Tagging Using Parallel Corpora: A Preliminary Investigation.," *Special Interest Group in Lexical Semantics (SIGLEX) Workshop, Association for Computational Linguistics,*, 2000.

[88] H.-C. Yang and C.-H. Lee, "Automatic Category Generation for Text Documents by Self-organizing Maps.," *0-7695-0619-4/00,IEEE,*, 2000.

[89] S. Eyassu and B. Gamback, "Classifying Amharic News Text Using Self-Organizing Maps," *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Ann Arbor*, p. 7178, 2005.

[90] P. Li and I. Farkas, "A Self-Organizing Connectionist Model of Bilingual Processing," *Elsevier Science BV All rights reserved*, 2002.

[91] A. Nikolaos and I. Helen, "Cross-language information retrieval using latent semantic indexing and self-organizing maps ," *Proceedings of International Joint Conference on Neural Networks, Budapest, Hungary.*, pp. 25–29, 2004.

[92] H. Demuth, M. Beale, and M. Hagan, "Neural Network Toolbox," *The MathWorks*, vol. Version 5.0, 2006.

[93] A. D. Kulkarni and G. M. Whitson, "Self Organizing Neural Networks With A Split/Merge Algorithm ," *ACM 089791-347-7/90/0003/0255*, 1990.

[94] S. Haykin, "Neural networks - a comprehensive foundation," *Prentice Hall International, Inc., Upper Saddle River, N.J*, p. 81, 1999.

[95] C. Lau, "Neural Networks Theoretical Foundations and Analysis," *IEEE PRESS, New Yprk*, pp. 74–90, 1992.

[96] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research 5*, 2004.

[97] T. Kohonen, K. Makisara, and T. Saramaki, "Phonotopic maps insightful representation of phonological features for speech recognition," *In Proc. of 7ICPR, International Conference on Pattern Recognition.Los Alamitos, CA. IEEE Computer Soc. Press*, pp. 182–185, 1984.

[98] S. Kaski and T. Kohonen, "Winner-take-all networks for physiological models of competitive learning," *Neural Networks*, vol. 7:, pp. 973–984, 1994.

[99] Kohonen, T., "Self-Organization and Associative Memory," *Springer-Verlag Berlin Heidelberg*, vol. Third Edition, 1989.

[100] R. Reilly and P. Tchimev, "Neural Network Approach to Solving the Traveling Salesman Problem," *JCSC*, pp. 41–61, 2003.

[101] L. Xia, D. Soergel, and Marchloninl, "Self-Organizing Semantic Map for Information Retrieval," *ACM 0-89791-448-1/91/0009/0262...1.50*, 1991.

[102] A. Ultsch, "Self organized feature planes for monitoring and knowledge acquisition of a chemical process.," *The International Conference on Artificial Neural Networks, Springer-Verlag, London:*, pp. 864–867, 1993.

[103] M. Oja, S. Kaski, and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum (2002) ," *Citeseer.IST*, 2003.

[104] J. Hollmen, "User profiling and classification for fraud detection in mobile communications networks ," *Thesis*, 2002.

[105] D. R. Chen, R. F. Hang, and L. Huang, "Breast Cancer Diagnosis Using Self-Organizing Map For Sonography," *Ultrasound in Medicine and Biology*, vol. 26, pp. 405–11, 2000.

[106] R. D. Lawrence, G. S. Almasi, and H. E. Rushmeier, "A scalable parallel algorithm for self-organizing maps with applications to sparse data problems.," *Data Mining Knowl. Discovery.*, vol. 3, no. 2, p. 171195, 1999.

[107] A. style, "Semitic languages," *Encyclopaedia Britannica.*, 2007. http://www.britannica.com/eb/article-9066720(Accessed: 12/07/2007).

[108] N. Fahim, H.and Abdel Baki, "Egyptian Demographic Center," ., 2000. http://www.frcu.eun.eg/www/homepage/cdc/cdc.htm. (Accessed: 25/04/2007).

[109] M. Salameh, "Quest for Middle East Oil: the US Versus the Asia Pacific Region," *Energy Policy*, vol. 31, no. (11), pp. 1085–1091, 2003.

[110] H. Cunningham, D. Maynard, K. Bontcheva, and T. V., "Gate: An Architecture for Development of Robust HLT Applications.," *In Proc. of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02), Philadelphia,*, vol. pp. 168-175, 2002.

[111] D. Maynard, K. Bontcheva, and H. Cunningham, "Automatic Language-Independent Induction of Gazetteer Lists.," *In Proceeding of 4th Language Resources and Evaluation Conference (LREC'04), Lisbon, Portual.*, 2004.

[112] M. Silberztein, "Nooj: a Linguistic Annotation System for Corpus Processing.," *In Proc. Association for Computational Linguis-*

tics of HLT/EMNLP, Vancouver, Canad, pp. pp. 10–11, 2005. http://www.aclweb.org/anthology/H/H05/H05-2006(Accessed: 23/04/2007).

[113] T. BuckWalter, "Arabic Morphological Analyser," *Linguistic Data Consortium(LDC), QAMUS, Suite 810, Philadelphia, PA, 19104-2653, USA*, 2002.

[114] A. Al-Hamlawi, "س ـدًا العرف في فن الصرف," *Dar Hera'a, Saudi Arabia*, 2003.

[115] I. Al-Sughaiyer and A. Al-Kharashi, I., "Arabic Morphological Analysis Techniques: A Comprehensive Survey," *Journal of the American Society of Information Science and Technology*, vol. 55(3), 2004.

[116] M. Maamouri, T. Buckwalter, and C. Cieri, "Dialectal arabic telephone speech corpus: Principles, tool design, and transcription conventions," *Paper template for Coling , Geneva*, 2004. http://papers.ldc.upenn.edu/NEMLAR2004/Dialectal-Arabic-telephone-speech-corpus.pdf (Accessed: 14/06/2006).

[117] S. L. Larkey, L. Ballesteros, and E. M. Connell, "Light stemming for arabic information retrieval," *Univ. of Massachusetts*, 2006.

[118] S. L. Larkey, L. Ballesteros, and E. M. Connell, "Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis," *SIGIR02, August 11-15, 2002, Tampere, Finland. ACM 1-58113-561-0/02/0008*, pp. 275–282, 2002.

[119] G. M. Wickens, "Arabic Grammar," *Cambridge, Cambridge University Press*, 1980.

[120] K. Darwish, "Probabilistic Methods for Searching OCR-Degraded Arabic Text," *PhD thesis, University of Maryland, Maryland, USA.*, 2003.

[121] F. Sadat and N. Habash, "Combination of Arabic Preprocessing Schemes for Statistical Machine Translation," *In Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL,,* pp. 1–8, 2006.

[122] R. Sproat, "Morphology and Commutation," *MIT Press, Cambridge, MA*, 1992.

[123] R. Huddleston, " Introduction to the Grammar of English.," *Cambridge, Cambridge University Press*, 1984.

[124] J. M. Sinclair, "A Way with Common Words," *In(Eds.)Hasselgard, Hilde. and Oksefjell, Signe. Oit of Corpora: Studies in Honour of Stig Johanssson. Amsterdam: Rodopi,,* pp. 157–180., 1999.

[125] E. C. M. G. Badawi and G. A., "Modern Written Arabic - A comprehensive Grammar," *London, Routledge*, 2004.

[126] V. M. Mol and H. Paulussen, "Natural Language Processing and Arabic: the Leuven Tandem Approach," *JEP-TALN, Arabic Language Processing, Fez,,* pp. 19–22., 2004.

[127] A. C. Say, S. Demir, O. Cetinoglu, and F. GN, "A Natural Language Processing Infrastructure for Turkish," *International Conference On Computational Linguistics, Proceedings of the 20th international conference on Computational Linguistics.*, 2004.

[128] J. Barnett, K. Knight, I. Mani, and E. Rich, "Knowledge and Natural Language Processing," *Communication of the ACM*, vol. 33, no. 8, 1990.

[129] Y. S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan, "Language Model Based Arabic Word Segmentation," *In Proc. of the 41st Annual Meeting*

on Association for Computational Linguistics. Sapporo, Japan., pp. 399–406, 2003.

[130] K. R. Beesley, "Arabci Finite-state Morphological Analysis and Generation," *In Proc. of the 16th Conference on Computational Linguistics. Copenhagen, Denemark,*, pp. 89–94, 1999.

[131] M. M. Syiam, Z. T. Fayed, and M. B. Habib, "An Intelligent System for Arabic Text Categorization," *IJICIS*, vol. 6, no. 1, 2006.

[132] S. Abuleil, K. Alsamara, and M. Evens, "Acquisition System for Arabic Noun Morphology," *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages,Philadelphia, Pennsylvania*, 2002.

[133] K. Taghva, R. Elkhoury, and J. Coombs, "Arabic Stemming Without A Root Dictionary," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC)*, vol. 1, pp. 152–157, 2005.

[134] K. Beesley, "Xerox Arabic Morphological Analyser Surface-Languagc," *Available from World Wide Web: (Unicode) Documentation*, 2003. http://www.xrce.xerox.com/competencies/content-analysis/arabic-inxight/arabic-surf-langunicode.pdf.(Accessed: 20/12/2007).

[135] K. Beesley, "Finite-State morphological analysis and generation of Arabic at Xerox research: status and plans in 2001," *Xerox Research Centre Europe*, 2001. http://www.elsnet.org/acl2001-arabic.html(Accessed: 20/12/2007).

[136] G. W. Miller, "Sakhr's Morphological analyser ," *Skakhr Company*, . http://www.sakhr.com/Sakhr_e/Technology/Morphology.htm?Indcx = 5& Main = Technology&Sub=Morphology (Accessed: 23/04/2007).

[137] E. Othman, K. Shaalan, and A. Rafea, "A Chart Parser for Analyzing Modern Standard Arabic," *Computa-*

tional Linguistics, . http://www.cs.cmu.edu/ãlavie/Sem-MT-wshp/Othman+Shaalan+Rafea_paper.pdf (Accessed: 23/04/2007).

[138] M. Porter, "Porter stemmer in Java," *An algorithm for suffix stripping.*, vol. 14, no. 3, pp. 130–137, 1980.

[139] P. N. Hilfinger, "Programming Languages and Compilers Class Notes #2: Lexica.," *Spring, CS 164,*, 2005.

[140] C. Fox, "A Stop List for General Text," *ACM-SIGIR.*, vol. 1-2,, no. 24, pp. 19-35, 1989.

[141] K. Darwish and D. W. O., "Term Selection for Searching Printed Arabic," *SIGIR02, ACM 1-58113-561-0/02/0008.Tampere, Finland.*, 2002.

[142] J. B. Lovins, "Development of a stemming algorithm," *Mechanical Translation and Computational Linguistics*, no. 11, pp. 22–31, 1968.

[143] R. Krovetz, "Viewing morphology as an inference process," *In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,*, pp. 191–202, 1993.

[144] C. D. Paice, "Another stemmer," *ACM SIGIR Forum archive*, vol. 24, no. 3, pp. 56–61, 1990.

[145] J. Xu and B. Croft, "Corpus-Based Stemming using Co-occurrence of Word Variants.," *ACM Transactions on Information Systems,*, vol. 16, no. 1, p. 6181, 1998.

[146] J. Sinclair, "Preliminary recommendations on text typology. ," *Eagles Document EAGTCWG-TTYP/P,*, 1996.

[147] S. Sharoff, "Towards basic categories for describing properties of texts in a corpus.," *In Proc. of Language Resources and Evaluation Conference (LREC04),Lisbon, Portugal,*, vol. V, pp. 1743–1746, 2004. http://www.ilc.cnr.it/EAGLES96/texttype/texttyp.html(Accessed: 11/11/2007).

[148] LDC, "Multiple-Translation Arabic (MTA) Part 2," *catalog number LDC2005T05 and ISBN 1-58563-328-3.*, 2003. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T05 (Accessed: 20/10/2007).

[149] J. Vesanto, J. Himberg, E. Al-honiemi, and J. Parhankangas, "Self-organizing map in Matlab: the SOM Toolbox," *Proceeding of the Matlab DSP Conference 1999, Espoo,Finland.*, pp. 35–40, 1999.

[150] H.-C. Yang and C.-H. Lee, "Towards Multilingual Information Discovery through a SOM Based Test Mining Approach," *Systems, Man and Cybernetics, IEEE International Conference*, vol. 5, 2002.

[151] M. G. Foody, "Status of land cover classification accuracy assessment," *Elsevier Science Inc. Remote Sensing of Environment 80. All rights reserved.*, p. 185201, 2002.

[152] K. A. Olsen, "Data visualization,," *BookRags. Retrieved from the World Wide Web*, 2002. http://www.bookrags.com/sciences/computerscience/data-visualization-csci03.html(Accessed: 24/04/2007).

[153] A. Flexer, "On the use of self-organizing maps for clustering and visualization.," *Intelligent Data Analysis*, vol. 5(5):, pp. 373–384, 2001.

[154] M. Maamouri and C. Cieri, "Resources for Arabic Natural Language Processing at the linguistic Data Consortium," *In Proceedings of the International Symposium on: The Processing of Arabic, Tunisia,*, pp. 125–146, 2002.

[155] L. Al-Sulaiti, "International Corpus of Arabic (ICA)," *School of Computing at the University of Leeds*, 2006. http://www.comp.leeds.ac.uk/eric/latifa/research.htm(Accessed: 20/10/2007).

[156] K. Papineni, S. Roukos, R. Ward, and W. Zhu, "Blue: a Method for Automatic Evaluation on Machine Translation," *Proceedings of the 40 Annual Meeting of the ACL.*, pp. 311–318, 2002.

[157] P. Nguyen, J. Gao, and M. Mahajan, "MSRLM: a scalable language modeling toolkit," *Microsoft Corporation*, 2007.

[158] A. S. Company, "ATA Machine Translation Software ," *ATA Software Technology Limited - London*, 1992.

[159] Wikipedia, "Standard deviation," ., 2007. http://en.wikipedia.org/wiki/Standard_deviation(Accessed: 20/10/2007).

[160] F. Lancaster, "Vocabulary control for information retrieval (Second ed.)," *Information Resources Press*, 1986.

[161] I. H. Witten and E. Frank, "WEKA Machine Learning Algorithms in Java," *Morgan Kaufmann Publishers*, pp. 265–321, 2000.

[162] A. Al-Marghilani, H. Zedan, and A. Ayesh, "Practical Approach Using Self-Organizing Maps For Multilingual Text Mining," *In Proc. EasyChair - Saudi innovation conference. Newcastle, UK*, 2007.

[163] A. Flexer, "Limitations of Self-Organizing Maps for Vector Quantization and Multidiminsional Scaling," *Technical Report Oefai-tr-96-23*, 2007.

[164] A. Rauber, d. Merkl, and M. Dittenbach, "The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, 2002.

Appendices

Appendix A

Buckwalter Transliterating System

Arabic script	Buckwalter	Arabic script	Buckwalter
ا	A	م	m
ب	b	ن	n
ت	t	ه‍ـ	h
ت‍ـ	v	و	w
ج	j	ي	y
ح	H	ى	Y
ه‍ـ	x	ة	p
د	d	fatHateen	F
د‍ـ	*	Dammateen	N
ر	r	kasrateen	K
ز	z	fatHa	a
س	s	Damma	u
س‍ـ	$	Kasra	i
ص	S	Shadda	
ض	D	Sukuun	o
ط	T	آ	|
ظ	Z	أ	>
ع	E	إ	<
غ	g	أو	&
ف	f	أي	}
ق	q	أ	i
ك	k	taTwiil	_
ل	l		

Appendix B

Stop-Words

This appendix shows Arabic-English stop-words

Arabic Stop-Words

,كذلك ,تلك ,وكَان ,علَى ,أَحد ,وليس ,به ,يكون ,وهو ,حتَى ,من ,في ,الَى ,يلي ,ضد ,بعد ,ان
,ليسب ,لَا ,ومن ,حين ,أَمَا ,الذي ,منذ ,ليس ,مسَاء ,عن ,لكن ,وعلَى ,إِن ,عليهَا ,فيهَا ,وبين ,التى
,تكون ,أَنه ,هذه ,ثم ,فقط ,وَالتي ,هذَا ,له ,ولكن ,لكنه ,مع ,دون ,حول ,عنه ,مَا ,أَي ,وكَانت
,بد ,كل ,الّذين ,عند ,لو ,ذلك ,فيه ,فئِن ,ه٤ولَاء ,لم ,اليوم ,لأَن ,لهم ,كَان ,نحو ,لن ,جدًا ,بين ,قد
,كمًا ,عليه ,علي ,إِذ ,أَو ,لهَا ,تحت ,فهو ,وفي ,بهَا ,منه ,عنهَا ,هو ,بل ,فقد ,ومع ,أَن ,وثي ,لدَى
,او ,اذَا ,هي ,حيث ,هل ,إِذَا ,إِلَى ,منهَا ,يوم ,معه ,قبل ,هنَاك ,أَمَام ,لذلك ,كَانت ,وقد ,هنَا ,كيف
,ظل ,اضحَى ,أَضحَى ,أَمسَى ,أَصبح ,اصبح ,مَايزَال ,لَايزَال ,لَازَال ,مَازَال ,إِلي ,الي ,لَا ,ومَا
,ضمن ,الحَالي ,ولَايزَال ,لَاسيمَا ,لعل ,ليت ,كَأَن ,إِن ,ليس ,صَار ,بَات ,مَاَانفك ,مَافت٤ي ,مَابرح
,ستكون ,فكَان ,الَا ,لهذَا ,وهذَا ,وَالذي ,وَان ,فَانه ,الذين ,انه ,اليهَا ,بدلَا ,اي ,ذَات ,وله ,اول
.الذَى ,هن ,الذي ,آل ,وأَبو ,وهي ,وأَن ,لدي ,بهذَا ,يمكن ,اليه ,الذي ,بئِن ,أَبو ,مَّا

English Stop-Words

a, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain't, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, amoungst, amount, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, aren't, around, as, aside, ask, asking, associated, at, available, away, awfully, back, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, bill, both, bottom, brief, but, by, c'mon, c's, call, came, can, can't, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, con, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldn't, course, cry, currently, definitely, described, despite, did, didn't, different, de, describe, detail, do, does, doesn't, doing, don't, done, down, downwards, during, each, edu, eg, eight, either, else, elsewhere, eleven, else, elsewhere, empty, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except, far, few, fifth, fifteen, fify, fill, find, fire, first, five, followed, following, follows, for, former, formerly, forth, four, forty, found, from, front, full, further, furthermore, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings, had, hadn't, happens, hardly, has, hasn't, have, haven't, having, he, he's, hello, help, hence, her, here, here's, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however, hundred, I, i'd, i'll, i'm, i've, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isn't, it, it'd, it'll, it's, its, itself, just, keep, keeps, kept, know, knows, known, last, lately, later, latter, latterly, least, less, lest, let, let's, like, liked, likely, little, look, looking, looks, ltd, made, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself, name,

namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, non, none, noone, nor, normally, not, nothing, novel, now, nowhere, obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, own, particular, particularly, per, perhaps, placed, please, plus, possible, presumably, probably, provides, que, quite, qv, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldn't, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure, t's, take, taken, tell, tends, th, than, thank, thanks, thanx, that, that's, thats, the, their, theirs, them, themselves, then, thence, there, there's, thereafter, thereby, therefore, therein, theres, thereupon, these, they, they'd, they'll, they're, they've, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually, value, various, very, via, viz, vs, want, wants, was, wasn't, way, we, we'd, we'll, we're, we've, welcome, well, went, were, weren't, what, what's, whatever, when, whence, whenever, where, where's, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, who's, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, won't, wonder, would, would, wouldn't, yes, yet, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves, zero

Appendix C

Samples of Existing Corpora

(Linguistic Data Consortium (LDC))

Sample 1 for Arabic

<seg id=1><hl><"DOC docid="AFA20030101.5900>
عزة ابراهيم يستقبل مسؤولا اقتصاديا سعوديا في بغداد <seg id=2><p><hl></seg>
بغداد ١-١ (الف ب).
ذكرت وكالة الانباء العراقية الرسمية ان نائب رئيس مجلس قيادة الثورة في العراق عزة ابراهيم استقبل اليوم الاربعاء في بغداد رئيس مجلس ادارة المركز السعودي لتطوير الصادرات عبد الرحمن الزامل. <p><p/><seg/>
<seg id=3>

وقالت الوكالة ان ابراهيم رحب في المناسبة بمستوى التعاون والتبادل التجاري بين العراق والسعودية. <seg/>
<seg id=4><p><p/>

واشارت الوكالة الى ان وزير التجارة العراقي محمد مهدي صالح شارك في اللقاء. <p><p/><seg/><seg
id=5>

وبدأت بغداد والرياض اللتان قطعتا علاقاتهما الدبلوماسية منذ حرب الخليج في ١٩٩١، عملية تقارب خلال قمة بيروت في آذار/مارس الماضي. <p><p/><seg/><seg id=6>
ونقلت السعودية هذا الاسبوع معلومات نشرت في صحيفة "نيويورك تايمز" الاميركية وفيها ان السعودية وافقت على وضع منشآتها العسكرية في تصرف الولايات المتحدة، في حال حصول حرب مع العراق. <p><p/><seg/>
<seg id=7>

وقال الامير عبد الرحمن بن عبد العزيز نائب وزير الدفاع السعودي في تصريح لصحيفة "عكاظ" السعودية ان "ما ادعته الصحيفة كلام غير صحيح". <seg/><p><p/><seg id=8>
واضاف ان "موقف المملكة واضح من البداية تجاه هذا الامر ونحن لا يمكن ان نضع مجالنا الجوي وقواعدنا بتصرف" الاميركيين. <seg/><p><p/><seg id=9>

وشنت السعودية قاعدة تنقوات الاميركية خلال حرب الخليج (كانون الثاني/يناير وشباط/فبراير ١٩٩١). <seg/>
<DOC/><p/>

Sample 2 for Arabic

<seg id=1><ll><"DOC docid="AFA20030111.4300>
الفرق الدولية تقوم بعمليات تفتيش جديدة في العراق </seg><p><ll/><seg id=2>
بغداد ١١-١ (اف ب) -
بدأت الفرق دولية اليوم السبت عمليات تفتيش جديدة لمواقع مشتبه بها في العراق، على ما افاد المركز الصحافي التابع لوزارة الاعلام العراقية. </seg><p><p/><seg id=3>

وعلى المفتشين الذين دخلت عمليتهم اسبوعها السابع تحديد ما اذا كان العراق يملك او يطور اسلحة دمار شامل كيميائية او بيولوجية او نووية او صواريخ بعيدة المدى ما يمثل انتهاكا لقرارات الامم المتحدة. </seg><p/>
<seg id=4><p>

وعاد فريق متخصص في الاسلحة البيولوجية الى مركزي تخزين في بغداد كان زارهما امس هما الدبش والعديل.
</seg><p><p/><seg id=5>

واتجه فريق من المختصين في الاسلحة الكيميائية الى الشمال الى وجهة غير معلومة. بينما اتجه فريق من المتخصصين في السلاح النووي الى موقع الفلوجة غرب بغداد. </seg><p><p/><seg id=6>

وزار فريق رابع من المتخصصين في الصواريخ موقع ابن سينا الكيميائي الذي سبق ان تمت زيارته في كانون الاول/ديسمبر. </seg><p><p/><seg id=7>

واليوم السبت هو اليوم الثالث والاربعون لعمليات التفتيش التي استؤنفت في ٢٧ تشرين الثاني/نوفمبر بعد اربع سنوات من توقفها. </seg><p><p/><seg id=8>

وصرح كبير المفتشين الدوليين هانس بليكس الخميس في مجلس الامن في نيويورك ان الخبراء "لم يجدوا دليلا ملتما يتيح تجريم" العراق بعد ستة اسابيع من عمليات التفتيش شملت ١٢٧ موقعا </seg><p/><DOC>

Sample 3 Translation in English

<DOC docid="AFA20030101.5900" sysid="ahd"><hl><seg id=1>
Izzet Ibrahim Meets Saudi Trade Official in Baghdad </seg></hl> <seg id=2>
Baghdad 1-1 (AFP) -
Iraq's official news agency reported that the Deputy Chairman of the Iraqi Revolutionary Command Council, Izzet Ibrahim, today met with Abdul Raluman al-Zamil, Managing Director of the Saudi Center for Export Development. </seg>
<seg id=3>
The agency said Ibrahim welcomed this occasion for trade exchange and cooperation between Iraq and Saudi Arabia. </seg> <seg id=4>
The agency also reported that the Iraqi Minister of Trade, Mohamed Mehdi Salih, took part in the meeting. </seg> <seg id=5>
Baghdad and Riyadh, who broke diplomatic relations during the Gulf War in 1991, began to improve their relations over the course of the Beirut Summit last March.
</seg> <seg id=6>
Saudi sources this week denied reports in the American New York Times that Saudi Arabia had agreed to allow the United States to use Saudi military bases should a war against Iraq take place. </seg> <seg id=7>
Prince Abdulrahman bin Azziz, Saudi Arabia's Defense Minister, said in a statement to Saudi newspaper Uqath that "what was claimed by the newspaper is incorrect." </seg> <seg id=8>
He added: "The Kingdom's position has been clear since the beginning and we are not able to place our airspace and bases" under American control.
</seg> <seg id=9> Saudi Arabia was a base for American forces during the Gulf War (January and February 1991). </seg> </DOC>

Sample 4 Translation in English

<DOC docid="AFA20030111.4300"
sysid="ahd"><hl><seg id=1> International Teams Carry out New Inspections in Iraq </seg></hl> <seg id=2> Baghdad 1-11 (AFP) -
The press center of the Iraqi Ministry of Information said that international teams have commenced new inspections of suspected sites in Iraq. </seg> <seg id=3>
The inspectors' role, in their 7th week of inspections, is to establish whether Iraq possess or is developing chemical, biological or nuclear weapons of mass destruction or long-range missiles which would constitute a breach of United Nations resolutions. </seg> <seg id=4>
A team of biological weapons experts returned to the two storage centers in Baghdad, Aldabsh and Aladil, which they had already visited yesterday. </seg>
<seg id=5>
A group of chemical weapons experts headed for the north to an unknown destination while nuclear weapons experts headed toward the Faluja site east of Baghdad. </seg> <seg id=6>
A fourth team of missile experts visited the Ibn Sina site which had already been inspected in December. </seg> <seg id=7>
Today, Saturday, is the 43rd day of inspection operations which started on November 27 after being stopped for four years. </seg> <seg id=8>
Hans Blix, the head of the international weapons inspection team, told the Security Council in New York that the experts, after six weeks of inspections covering 127 sites, had "found no convincing evidence to incriminate Iraq." </seg> </DOC>

Appendix D

International Corpus of Arabic (ICA)

Sample


```
?>xml version="1.0" encoding="utf-8<? "
>tei.2<
>teiHeader id="EduO3<"
>fileDesc<
>titleStmt<
>title/>التعليم في العراق <title<
>author/>    أحمد أبوزيد محمد <author<
>respStmt><resp>compiled by</resp<
>name>Latifa Al-Sulaiti</name></respStmt<
/>titleStmt<
>publicationStmt<
>publisher> Ministry of education, Saudi Arabia </publisher<
>pubPlace>Saudi Arabia </pubPlace<
>date>2003</date<
/>publicationStmt<
>sourceDesc<
>p>created in machine-readable form in http://www.almarefah.com/</p<
/>sourceDesc<
/>fileDesc<
>encodingDesc<
>projectDesc<
>p>Texts collected for use in the
    Corpus of Contemporary Arabic project, June, 2003</p<
/>projectDesc<
>samplingDecl<
>p>Whole text of 4020 words copied from the site</p<
/>samplingDecl<
/>encodingDesc<
>profileDesc<
>creation<
>date value="2003-01">Jan 2003,Issue no 106 </date<
>rs type="city">Riyadh, Saudi Arabia</rs<
/>creation<
>langUsage>Arabic</langUsage<
>textClass<
>textDesc ana="Education<"
>channel mode="w">print; written</channel<
>constitution type="single</"
>derivation type="original</"
>domain type="socsci</"
>factuality type="fact</"
>interaction type="none" active="singular</"
```

>firstLang>Arabic</firstLang<
>langKnown>Unknown</langKnown<
>residence>Unknown</residence<
>education>Unknown</education<
>occupation>Unknown</occupation<
/>person<
/>particDesc<
/>textClass<
/>profileDesc<
/>teiHeader<
>text<
>body<

كان النظام التعليمي في العراق من أكثر النظم تقدمًا في العالم العربي قبل عام ١٩٩٠. بيد أن هذا النظام تدهور تدهورًا كبيرًا نتيجة الحروب التي دخلها النظام السابق وما أعقبها من فرض عقوبات دولية على البلاد، فضلًا عما أدخلها في دائرة الإهمال والانعزال وأورث مشكلات ضخمة ما زالت البلاد تعانيها في الوقت الحالي. وقد تفاقمت الأوضاع نتيجة أعمال التدمير والنهب والتعطيل لمؤسسات الدولة، والتي وقعت عقب شهر مارس ٢٠٠٣م في أعقاب سقوط العاصمة بغداد وانهيار النظام الصدامي ودخول القوات الأمريكية والبريطانية للبلاد. ويأمل المجتمع الدولي بعد أن يستتب الأمر في العراق ويزول الحكم وتطالب السلطة الحكومة وطنية عراقية منتخبة أن يتحرك العراق بسرعة لإعادة بناء النظام التعليمي وتأهيله وتجديده. وهناك أسباب كثيرة تدعونا إلى التفاؤل. أولها ما يتمتع به العراق من تاريخ تفاعل عريق يمتد عبر قرون من الزمن، وبعد جذوره في الحقبة التي تصدر بها العلماء العرب في العالم، وفي موضوعات متعددة مثل الرياضيات والطب. والسبب الثاني أن العراق كان يتمتع قبل عقد مضى حالة من الانتعاش الكبير في مجال التعليم، إذ تدفقت الأموال الأجواء، ولكنها أعقبتها. والسبب الثالث أن الحكومة التي كانت تستعدي التبادل الحر للأفكار والمعرفة قد لفظ الإطاحة بها. والسبب الرابع لهذا التفاؤل يكمن فيما يمتلكه العراق من موارد الثروة، وليس من المستبعد أن يقوم ثانية باستخدامها للأغراض التعليمية بعد أن مر بتجارب الحروب والتزامات المديونية. وهذه الموارد الهائلة يجب ألا تبدد على الآلة والمغامرات العسكرية بل ينبغي تسخيرها لتحسين حياة الشعب العراقي. فقد حان وقت النهوض للتطور العراق الفكري والثقافي. ومما لا شك فيه أن كلًا من وزارة التربية والتعليم ووزارة التعليم العالي والبحث العلمي سيواجهان قريبًا تحديًا للعودة إلى الظروف الطبيعية في مرحلة ما بعد الحرب، وبالتالي إعادة البناء التدريجي وتجديد نظام التعليم بكامله على المستوى الوطني. وهذه المهمة ستكون أقل كلفة في القطاعات الثلاث في مجال العراق حيث تعرض نظام التعليم إلى أضرار أقل في مرافق البنية التحتية وخدمات التعليم، وحيث تولت منظمة اليونيسكو مع صندوق اليونيسيف لتنفيذ برنامج التعليم هناك بصورة مشتركة، فشهدت المؤسسات التعليمية تطورًا كبيرًا، كما تم تزويدها بالمواد التعليمية على جميع مستويات التعليم وزادت قدرة الأفراد على الوصول إلى التعليم. وقد طرأ تطور على المرافق التعليمية بسبب توفر عنصر التمويل النفطي للصرف على عملية البناء والثراء عليًا، بينما اختلف الأمر في منطقتي وسط وجنوب العراق، حيث كان تزويد المعدات يتم من طريق الأمم المتحدة في إطار برنامج «النفط مقابل الغذاء»، وذلك بعد أن حالة معظم المدارس في منطقتي الوسط والجنوب كانت متدنية بسبب النقص في المعدات التي يحتاج إليها قطاع التعليم وسيطرة الحكومة العراقية السابقة على تنفيذ البرنامج في ١٥ محافظة تغطي منطقتي الوسط والجنوب مع دور رئاسي فقط للأمم المتحدة في توزيع المواد المشتراة. وعلى أية حكومة عراقية مركزية قادمة أن تسعى لسد الفجوة القائمة في قطاع التعليم في محافظات العراق الثمانية عشرة سواء من حيث البنية التحتية أو التأثيث أو توفير المعلمين والكتب والمستلزمات المدرسية. ونتوقع أن يتسم جهد المسؤولين في البناء على تطوير النظام التعليمي من حيث المناهج وتوفير العوامل والمؤسسات التعليمية المختلفة، بل وإعادة بناء بعضها، وتوسيع رقعة المستفيدين، ودعم هيكل المتعلق بالهيكل التعليمي، وتعريف ما لدى العراقيين من متابعة ومواكبة للخطة العلمية في الدول المتقدمة. وسيعتبر النظام التعليمي القائم هو الأمر في العراق - والذي يشابه كثيرًا مثيله في معظم الدول العربية - والذي نتوقع أن لا يتغير في هيكله العام في المدى القريب استقراره منذ زمن طويل وشبيه مع برامج التعليم العالمية التي تدعو لزيادة مرحلة التعليم الأساسي الإلزامي أطول فترة ممكنة الغربي وصحراء في الغرب ووادٍ خصب تتوسطه الفرات ودجلة. يحدها غربًا إيران، وغربًا سوريا والأردن، وشمالًا تركيا وجنوبًا السعودية والكويت. المصدر: موسوعة

/>body<
/>text<
/>tei.2<

Appendix E

Source Code

This appendix shows subset of the whole code.

Matlab Code

This appendix shows subset of the whole code.

Listing E.1: Preparation Data Code

```
1 function [docall , docs , wcdoc , lang_indcx , q] = som_gct_docs();
2 q = false;
3 load writedoc.txt;              % load the output file which is generated
4 [r c]= size(writedoc);          % show the size of output file
5 rr=1; cc=1;                     % initialize row and column with 1
6 if writedoc(rr)==88888888
7     lang_index(cc)=1;
8 elseif writedoc(rr) == 99999999
9         lang_index(cc) =2;
10 else
11     lang_index(cc) = 3;
12 end;
13 for i=2:r
14     if (writedoc(i) == 88888888) | (writedoc(i) == 99999999)
15         |(writcdoc(i) == 77777777)
16         cc=cc+1;
17         rr=1;
18         if writedoc(i)==88888888
19            lang_index(cc)=1;
20         elseif writedoc(i) == 99999999
21            lang_index(cc) = 2;
22         else
23             lang_index(cc) = 3;
24             q = true;
25         end;
26     else
27         docs(rr,cc)= writedoc(i);
28         rr = rr+1;
29    end;
30 end;
31 [n m] = size(docs);     %  n = number of words and m = number of docs
32 docall = [];            %  Create new empty matrix k x 1
33 found = 0;
34 for i=1:n
35     for j=1:m
```

```
36          if docs(i,j)==0
37              docs(i,j)=10000000;  % write 10000000 to the short vectors
38          end;
39          word=docs(i,j);
40              for k=1:size(docall)
41                  if word == docall(k,1)
42                      found = 1;
43                      break;
44                  end;
45                  found = 0;
46              end;
47          if found==0 && word~=10000000
48              docall = [docall; word];  % create a unique word matrix of
49                                        % documents all.
50          end;
51      end;
52  end;
53  docall = sort(docall,1,'ascend');     % sort the docall matrix
54  [N M] = size(docall);                 % N= number words and M = 1 (vector)
55  wcdoc  = zeros(N,m);                  % intialize the wcdoc matrix by zero
56  indexdoc = zeros(N,1);
57  for i=1:n
58      for j=1:m
59          for k=1:N
60              indexdoc(k) = k;
61              word = docs(i,j);
62              if word == docall(k,1)
63                  wcdoc(k,j)=wcdoc(k,j)+1; % create a matrix of N x m of word
64                                           % frequency ( number of word
65                                           % occurrences in a document)
66                  break;
67              end;
68          end;
69      end;
70  end;
```

Listing E.2: Training SOM.

```
1 echo off;
2 clear all;
3 clc
4 slCharacterEncoding('ISO_8859-1');
5 prep_data=1;
6 prep_data = input( 'Would_you_like_to_prepare_the_input_data:_Please_enter
7 ((1,_yes)_or_(0,_no):_');
8 if prep_data== 1
9   load som_get_dict.txt;          % load bilingual dictionary
10  [catg] = som_get_catg();
11  [docall, wcdoc, lang_index, q] = som_get_docs();
12                                  % call function get_documents
13  [R,C]=size(wcdoc);              % determind the size of wcdoc matrix
14  wcdoc = wcdoc';
15  Titles={};
16  Wordss={};
17  Temp = C;
18  C = R;
19  R = Temp;
20  fid=fopen('asma.data','w');     % save the structure data
21  fprintf(fid,'%d\n',C);
22  fprintf(fid,'\n\t');
23  for j=1:C
24      fprintf(fid,'W%d\t',j);     % write the label of words
25  end
26  fprintf(fid,'\n');
27  [get_dictr, get_dictc]=size(som_get_dict);
28  for i=1:R
29     TempMax = 0;
30     Tempos = 0;
31     for j=1:C
32        fprintf(fid,'\%d\t',wcdoc(i,j));
33        if wcdoc(i,j)>=TempMax
34           TempMax = wcdoc(i,j); % determine the max occurancy of words
35           Tempos = j;           % store the position of the word in
36                                 % docall temperorily
37        end;
38     end
39     MaxPos(i)=Tempos;          % determine the position of the max word
```

```
40      MaxVal(i)=TempMax;
41      catgword(i) = docall(MaxPos(i)); % get the word number from docall
42      wordfound = false;
43      for k=1:get_dictr
44          if catgword(i)==som_get_dict(k,1) % search the word number which has the
45                                            % highest repetition in the
46                                            % dictionary and get its category number
47              wordfound = true;
48              if lang_index(i)==1
49                  Titles(i)=catg(som_get_dict(k,2),2);
50                                            % determine the category number
51                                            % from dictionary
52                                            % if langaue_index equal 1 get the
53                                            % Arabic category
54              elseif lang_index(i)==2       % if langaue_index equal 2 get the
55                  Titles(i)=catg(som_get_dict(k,2),1); % English category
56              else
57                  qAE =1;
58                 if qAE==1
59                     Titles(i)={'Query'};  % if it is query
60                  else
61                    Titles(i)={'        '};
62                  end;
63              end;
64               fprintf(fid,'\%s',Titles{i});% add category into asma data
65               fprintf(fid,'[\%d]\n',i);
66              break;
67          end; % end if statement
68      end; % end search from dictionary database for a document
69      if ~wordfound
70         messagehalt ='The_word_used_in_one_of_the_document_was_not_found';
71         zzzz= input(messagehalt);
72         quit;
73         end;
74      fclose(fid);
75      echo off;
76      bar(MaxVal, 'stacked');
77      title('Max_occurancy_of_words_for_each_document');
78      ylabel('Max_Occurances'), xlabel('Documents')
79      elseif prep_data== 0
80            sDasma = som_read_data('asma.data');
81            sDasma = som_normalize_data(sDasma,'range');%'var');
```

```
82          sMap = som_make_struct(sDasma,'msize',[5  4]);
83          sMap = som_autolabel(sMap,sDasma,'add');       %'add''vote');
84          colormap(1-gray)
85          som_docs_show(sMap,'norm','n')
86          som_show_add('label',sMap,'Textsize',7,'TextColor','b','Subplot',2)
87          h = som_hits(sMap,sDasma);
88          som_show_add('hit',h,'MarkerColor','g','Subplot',1)
89          som_show_clear('hit',1)
90          [dlen dim] = size(sDasma.data);      % determine the size of sDasma
91          n_cats= 1;
92          cat_freq(n_cats) = 1;
93          cat_names(1)=sDasma.labels(1);       % save first category with the
94                                               % first of cate_names
95          finished = false;
96          while ((n_cats < dlen) && ~finished)
97                j = n_cats +1;
98                for i=j:dlen
99                    k = 1;
100                   found = false;
101               while ((k <= n_cats) && ~found)
102                   if ismember(sDasma.labels{i},cat_names{k})
103                                            % check the category name with
104                       cat_freq(k)=cat_freq(k)+1;
105                                            % cate_names in the categories file
106                       found = true;
107                   else
108                   k=k+1;
109                   end;
110               end;
111               if ~found
112                   n_cats = n_cats + 1;
113                   cat_freq(n_cats) = 1;
114                   cat_names(n_cats) = sDasma.labels(i);
115                end;
116               end;
117               if i >= dlen
118                    finished = true;
119               end;
120           end;
121          echo off
122          warning on
123    end;              % end preparing the input data
```

Java Code

Listing E.3: Multilingual Morphological Analysis

```
1  package MainApplication;
2  import javax.swing.*;
3  import java.awt.*;
4  import javax.swing.event.*;
5  import MainApplication.*;
6  import englishstemmer.*;
7  public class AppInterface extends javax.swing.JFrame {
8      /** Creates new form AppInterface */
9      public AppInterface() {
10         initComponents();
11     }
12     /** This method is called from within the constructor to
13      * initialize the form.
14      * WARNING: Do NOT modify this code. The content of this method is
15      * always regenerated by the Form Editor.
16      */
17     // <editor-fold defaultstate="collapsed" desc=" Generated Code ">
18     private void initComponents() {
19         jLabel1 = new javax.swing.JLabel();
20         jPanel1 = new javax.swing.JPanel();
21         ArabicFolderLabel = new javax.swing.JLabel();
22         ArbFolderButton = new javax.swing.JButton();
23         ArabicRemoveHTMLButton = new javax.swing.JButton();
24         jButton8 = new javax.swing.JButton();
25         jButton9 = new javax.swing.JButton();
26         jButton10 = new javax.swing.JButton();
27         jPanel2 = new javax.swing.JPanel();
28         EngFloderButton = new javax.swing.JButton();
29         GetWordsNumberButton = new javax.swing.JButton();
30         GetWordsRootsButton = new javax.swing.JButton();
31         EnglishRemoveHTMLButton = new javax.swing.JButton();
32         EnglishFolderLabel = new javax.swing.JLabel();
33         jLabel2 = new javax.swing.JLabel();
34         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
35         setTitle("<<Application Name Here>>");
36         setBackground(new java.awt.Color(204, 255, 204));
37         setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
```

```
38          setResizable(false);
39          jLabel1.setBackground(new java.awt.Color(204, 255, 255));
40          jLabel1.setFont(new java.awt.Font("Arial", 1, 18));
41          jLabel1.setForeground(new java.awt.Color(0, 0, 204));
42          jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
43          jLabel1.setText("Put Application Name Here");
44          jPanel1.setBackground(new java.awt.Color(255, 204, 204));
45          ArabicFolderLabel.setText("Arabic Folder");
46          ArbFolderButton.setText("Select Folder");
47          ArbFolderButton.addMouseListener(new java.awt.event.MouseAdapter() {
48              public void mouseClicked(java.awt.event.MouseEvent evt) {
49                  ArbFolderButtonMouseClicked(evt);
50              }
51          });
52          ArabicRemoveHTMLButton.setText("Remove HTML Tags");
53          jButton8.setText("Get Words Roots");
54          jButton9.setText("Remove Fade Words");
55          jButton10.setText("Get Words Numbers");
56          javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
57          jPanel1.setLayout(jPanel1Layout);
58          jPanel1Layout.setHorizontalGroup(
59              jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
60              .addGroup(jPanel1Layout.createSequentialGroup()
61                  .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
62                  .Alignment.LEADING)
63                      .addGroup(jPanel1Layout.createSequentialGroup()
64                          .addGap(20, 20, 20)
65                          .addGroup(jPanel1Layout.createParallelGroup(javax.swing
66                          .GroupLayout.Alignment.LEADING)
67                              .addComponent(ArbFolderButton, javax.swing.GroupLayout
68                              .DEFAULT_SIZE, 179, Short.MAX_VALUE)
69                              .addComponent(ArabicRemoveHTMLButton, javax.swing.GroupLayout
70                              .DEFAULT_SIZE, 179, Short.MAX_VALUE)
71                              .addComponent(jButton9, javax.swing.GroupLayout
72                              .DEFAULT_SIZE, 179, Short.MAX_VALUE)
73                              .addComponent(jButton8, javax.swing.GroupLayout
74                              .DEFAULT_SIZE, 179, Short.MAX_VALUE)
75                              .addComponent(jButton10, javax.swing.GroupLayout
76                              .DEFAULT_SIZE, 179, Short.MAX_VALUE)))
77                  .addGroup(jPanel1Layout.createSequentialGroup()
78                      .addContainerGap()
79                      .addComponent(ArabicFolderLabel, javax.swing.GroupLayout
```

```
80                          .PREFERRED_SIZE, 172, javax.swing.GroupLayout.PREFERRED_SIZE)))
81                  .addContainerGap())
82          );
83          jPanel1Layout.setVerticalGroup(
84              jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
85              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout
86              .createSequentialGroup()
87                  .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
88                  .addComponent(ArabicFolderLabel, javax.swing.GroupLayout
89                  .PREFERRED_SIZE, 28, javax.swing.GroupLayout.PREFERRED_SIZE)
90                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
91                  .addComponent(ArbFolderButton, javax.swing.GroupLayout
92                  .PREFERRED_SIZE, 31, javax.swing.GroupLayout.PREFERRED_SIZE)
93                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
94                  .addComponent(ArabicRemoveHTMLButton, javax.swing.GroupLayout
95                  .PREFERRED_SIZE, 31, javax.swing.GroupLayout.PREFERRED_SIZE)
96                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
97                  .addComponent(jButton9, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
98                  javax.swing.GroupLayout.PREFERRED_SIZE)
99                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
100                 .addComponent(jButton8, javax.swing.GroupLayout.PREFERRED_SIZE, 31,javax
101                 .swing.GroupLayout.PREFERRED_SIZE)
102                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
103                 .addComponent(jButton10, javax.swing.GroupLayout.PREFERRED_SIZE, 32,javax
104                 .swing.GroupLayout.PREFERRED_SIZE).addGap(27, 27, 27))
105         );
106         jPanel2.setBackground(new java.awt.Color(255, 255, 204));
107         EngFloderButton.setText("Select Folder");
108         EngFloderButton.addMouseListener(new java.awt.event.MouseAdapter() {
109             public void mouseClicked(java.awt.event.MouseEvent evt) {
110                 EngFloderButtonMouseClicked(evt);
111             }
112         });
113         GetWordsNumberButton.setText("Get Words Numbers");
114         GetWordsNumberButton.addMouseListener(new java.awt.event.MouseAdapter() {
115             public void mouseClicked(java.awt.event.MouseEvent evt) {
116                 GetWordsNumberButtonMouseClicked(evt);
117             }
118         });
119         GetWordsRootsButton.setText("Remove Fade Words & Get Roots");
120         GetWordsRootsButton.addMouseListener(new java.awt.event.MouseAdapter() {
121             public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
122                 GetWordsRootsButtonMouseClicked(evt);
123             }
124         });
125         EnglishRemoveHTMLButton.setText("Remove HIML Tags");
126         EnglishRemoveHTMLButton.addMouseListener(new java.awt.event.MouseAdapter() {
127             public void mouseClicked(java.awt.event.MouseEvent evt) {
128                 EnglishRemoveHTMLButtonMouseClicked(evt);
129             }
130         });
131         EnglishFolderLabel.setText("English Folder");
132         EnglishFolderLabel.setBorder(javax.swing.BorderFactory.createEtchedBorder
133         (new java.awt.Color(0, 51, 255), new java.awt.Color(0, 0, 255)));
134         javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
135         jPanel2.setLayout(jPanel2Layout);
136         jPanel2Layout.setHorizontalGroup(
137             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
138             .addGroup(jPanel2Layout.createSequentialGroup()
139                 .addContainerGap()
140                 .addComponent(EnglishRemoveHTMLButton, javax.swing.GroupLayout
141                 .DEFAULT_SIZE, 193, Short.MAX_VALUE).addGap(12, 12, 12))
142             .addGroup(jPanel2Layout.createSequentialGroup()
143                 .addContainerGap()
144                 .addComponent(GetWordsRootsButton, javax.swing.GroupLayout.DEFAULT_SIZE,
145                 javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
146                 .addGap(12, 12, 12))
147                 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout
148                 .createSequentialGroup()
149                 .addGap(12, 12, 12)
150                 .addComponent(GetWordsNumberButton, javax.swing.GroupLayout
151                 .DEFAULT_SIZE, 193, Short.MAX_VALUE)
152                 .addContainerGap())
153             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout
154             .createSequentialGroup()
155                 .addGap(12, 12, 12)
156                 .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
157                 .Alignment.TRAILING).addComponent(EnglishFolderLabel, javax.swing
158                 .GroupLayout.Alignment
159                     .LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 191, Short.MAX_VALUE)
160                     .addComponent(EngFloderButton, javax.swing.GroupLayout.Alignment
161                     .LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, 191, Short.MAX_VALUE))
162                 .addGap(12, 12, 12))
163         );
```

```
164         jPanel2Layout.setVerticalGroup(
165             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
166             .addGroup(jPanel2Layout.createSequentialGroup()
167                 .addContainerGap()
168                 .addComponent(EnglishFolderLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
169                  javax.swing.GroupLayout.PREFERRED_SIZE)
170                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
171                 .addComponent(EngFloderButton, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
172                  javax.swing.GroupLayout.PREFERRED_SIZE).addGap(14, 14, 14)
173                 .addComponent(EnglishRemoveHTMLButton, javax.swing.GroupLayout.PREFERRED_SIZE,
174                  31, javax.swing.GroupLayout.PREFERRED_SIZE).addGap(15, 15, 15)
175                 .addComponent(GetWordsRootsButton, javax.swing.GroupLayout.PREFERRED_SIZE,
176                 31, javax.swing.GroupLayout.PREFERRED_SIZE).addGap(15, 15, 15)
177                 .addComponent(GetWordsNumberButton, javax.swing.GroupLayout.DEFAULT_SIZE,
178                 32, Short.MAX_VALUE).addGap(37, 37, 37))
179         );
180         jLabel2.setFont(new java.awt.Font("Tahoma", 2, 8));
181         jLabel2.setText("This Application was Buillding By Abo Ahmad");
182         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
183         getContentPane().setLayout(layout);
184         layout.setHorizontalGroup(
185             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
186             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout
187             .createSequentialGroup().addContainerGap()
188                 .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 455,
189                  Short.MAX_VALUE)).addGroup(layout.createSequentialGroup()
190                 .addContainerGap()
191                 .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
192                 javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout
193                 .PREFERRED_SIZE).addGap(21, 21, 21)
194                 .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing
195                 .GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
196                 .addContainerGap())
197             .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 192, javax
198             .swing.GroupLayout.PREFERRED_SIZE)
199         );
200         layout.setVerticalGroup(
201             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
202             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout
203             .createSequentialGroup().addContainerGap()
204                 .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
205                 javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
200                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout
207                 .Alignment.TRAILING).addGroup(layout.createSequentialGroup()
208                         .addGap(20, 20, 20)
209                         .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
210                          javax.swing
211                             .GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
212                        .addGroup(layout.createSequentialGroup()
213                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement
214                             .RELATED)
215                             .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
216                              javax.swing
217                             .GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
218                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
219                     .addComponent(jLabel2))
220             );
221             pack();
222         }// </editor-fold>
223         private void GetWordsNumberButtonMouseClicked(java.awt.event.MouseEvent evt) {
    try{
224             WordsNumbers.readWordsWriteNumbers("read.txt");
225         }catch (Exception e){System.out.println(e.toString());}
226          JOptionPane.showMessageDialog(null,"Ok. <<See writeOut1.txt and writeOut2.txt Files>>");
227          }
228          private void GetWordsRootsButtonMouseClicked(java.awt.event.MouseEvent evt) {
229            Stemmer2.readFolder(EnglishFolderLabel.getText());
230            JOptionPane.showMessageDialog(null,
231            "Ok. The Files has been stemmered, <<See Read.txt file>>");
232          }
233          private void EnglishRemoveHTMLButtonMouseClicked(java.awt.event.MouseEvent evt) {
234           HTMLRemoveTags html=new HTMLRemoveTags();
235           try{
236           html.HTMLRemoveFolder(EnglishFolderLabel.getText());
237           }catch(Exception e){System.out.println(e.toString());}
238            JOptionPane.showMessageDialog(null,
239            "Ok. <<All Files in selected folder has been Parssing>>");
240          }
241          private void ArbFolderButtonMouseClicked(java.awt.event.MouseEvent evt) {
242           JFileChooser filechooser=new JFileChooser();
243           filechooser.showOpenDialog(this);
244           ArabicFolderLabel.setText(filechooser.getCurrentDirectory().getPath());
245          }
246          private void EngFloderButtonMouseClicked(java.awt.event.MouseEvent evt) {
```

```
247         JFileChooser filechooser=new JFileChooser();
248         filechooser.showOpenDialog(this);
249         EnglishFolderLabel.setText(filechooser.getCurrentDirectory().getPath());
250       }
251       /**
252        * @param args the command line arguments
253        */
254       public static void main(String args[]) {
255           java.awt.EventQueue.invokeLater(new Runnable() {
256               public void run() {
257                   new AppInterface().setVisible(true);
258               }
259           });
260       }
261       // Variables declaration - do not modify
262       private javax.swing.JLabel ArabicFolderLabel;
263       private javax.swing.JButton ArabicRemoveHTMLButton;
264       private javax.swing.JButton ArbFolderButton;
265       private javax.swing.JButton EngFloderButton;
266       private javax.swing.JLabel EnglishFolderLabel;
267       private javax.swing.JButton EnglishRemoveHTMLButton;
268       private javax.swing.JButton GetWordsNumberButton;
269       private javax.swing.JButton GetWordsRootsButton;
270       private javax.swing.JButton jButton10;
271       private javax.swing.JButton jButton8;
272       private javax.swing.JButton jButton9;
273       private javax.swing.JLabel jLabel1;
274       private javax.swing.JLabel jLabel2;
275       private javax.swing.JPanel jPanel1;
276       private javax.swing.JPanel jPanel2;
277       // End of variables declaration
278       }
```

Listing E.4: Arabic Stemmer Code

```
1 package arabicstemmer;
2 import java.io.*;
3 import java.util.StringTokenizer;
4 import java.sql.*;
5 public class GetEnglishStemmer {
6     public static void main(String[] args) throws Exception{
```

```
7         File edir = new File("arabicfiles");
8         String[] echildren = edir.list();
9         if (echildren == null) {
10                // Either dir does not exist or is not a directory
11                System.out.println("Either dir does not exist or is not a directory");
12        } else {
13              for (int i=0;i<echildren.length;i++) {
14
15                    File f = new File("arabicfiles\\" + echildren[i]);
16                    if (f.isFile()){
17                      //  readFiles(new File("arabicfiles\\" + echildren[i]));
18                        System.out.println("arabicfiles\\" + echildren[i]);
19                    }else{
20                        String[] subFile=f.list();
21                        for (int j=0;j<subFile.length;j++){
22                         System.out.println("arabicfiles\\" + echildren[i]+"\\"+subFile[j]);
23                        }
24                    }
25                }
26        }
27 }
28 public static void readFiles (File f) throws Exception {
29 GetArabicStemmer wa=new GetArabicStemmer();
30 int arabparagraph=0;
31 int englishparagraph=0;
32 String word="";
33 File inputFile = f;
34 File outputFile = new File("read.txt");
35 FileReader in = new FileReader(inputFile);
36 FileWriter out = new FileWriter(outputFile,true);
37 DataInputStream dis = new DataInputStream(System.in);
38 out.write("*");
39 out.write(13);
40 out.write(10);
41 int c;
42 //Store All File In S
43 word="";
44 while ((c = in.read()) != -1){
45     if (c==13){
46         out.write(13);
47         out.write(10);
48       }
```

```
49        if (c==32) {
50             if (!wa.isFadeWord(word)){
51               String root=wa.getRoot(word);
52               if (root!="")
53                    out.write(root+" ");
54             }
55               word="";
56        }else{
57             word=word+(char)c;
58        }
59 }
60     in.close();
61     out.close();
62     System.out.println("OK");
63 }
64 }
65 %Listing D.4: Arbic Stemmer Code\\
66 \line(1,0){432.0000}\\
67 {\scriptsize
68 \begin{lstlisting}
69 package arabicstemmer;
70 import databasc.DBAccess;
71 import englishstemmer.*;
72 import java.sql.*;
73 public class WordAnalyze {
74 private  ResultSet rs=null;
75 DBAccess db = new DBAccess("jdbc:odbc:Dictionary","sun.jdbc.odbc.JdbcOdbcDriver");
76     public String getRoot(String word) throws Exception{
77          String sword="";
78          rs=db.runSelect("select * from Derivation where derivative='"+word+"'");
79          int n=0;
80          if (rs.next()){
81               n=1;
82          }
83          if (n==1){
84               int i=rs.getInt("id");
85               rs=db.runSelect("select * from newtab where id=" + i);
86               rs.next();
87               sword=rs.getString("Arabic");
88          }
89          return sword;
90     }
```

```
91      public boolean isFadeWord(String word){
92          int n=0;
93          boolean test=false;
94          try{
95              rs = db.runSelect("select * from fades where word='"+word+"'");
96              if (rs.next()) {
97                  n = 1;
98              }
99          }catch (Exception e){System.out.println(e);}
100         if (n==1){
101            test=true;
102         }
103         return test;
104     }
105 }
```

Listing E.5: Database Access Code

```
1 package database;
2 import java.sql.*;
3 public class DBAccess  {
4 private String datasourse="";
5 private String driver;
6 private ResultSet rs=null;
7 Statement stmt;
8 Connection con;
9 public DBAccess(String db,String driv){
10   datasourse=db;
11   driver=driv;
12     try {
13         Class.forName(driver);
14       } catch(Exception ex) {
15         System.err.print("Exception: ");
16         System.err.println(ex.getMessage());
17       }
18 }
19 public ResultSet runSelect(String sqls){
20 try{
21                 String query = sqls;
22                 con = DriverManager.getConnection(datasourse);
23                 stmt = con.createStatement(
```

```
24                          ResultSet.TYPE_SCROLL_SENSITIVE,
25                          ResultSet.CONCUR_READ_ONLY);
26                  rs = stmt.executeQuery(query);
27 }
28                   catch(Exception ex) {
29                          System.err.print("Exception: ");
30                          System.err.println(ex.getMessage());
31                  }
32                  finally
33                  {
34                       return rs;
35                  }
36                  }
37 public void runSQL(String sql){
38   try{
39                  String query = sql;
40                  con = DriverManager.getConnection(datasourse);
41                  stmt = con.createStatement();
42                       stmt.executeUpdate(query);
43    }catch (Exception e){
44      System.out.println(e.toString());
45    }
46 }
47 public void disConnect(){
48 try{
49      stmt.close();
50      con.close();
51      }
52      catch (Exception e){
53      System.out.println(e.toString());
54    }
55 }
56 }
```