Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

IMPROVING THE INDEPENDENT COMPONENT ANALYSIS BASED ON METAHEURISTIC ALGORITHMS FOR TEXT CLUSTERING

A Dissertation

Submitted to the Council of the College of Information Technology, University of Babylon in Partial Fulfillment of the Requirements for the Doctor of Philosophy Degree in Information Technology-Software

Hafedh Ali Shabat Mizael

Supervised by

Prof. Dr. Nidaa Abdual Muhsin Abbas

2021A.D. **1443 A.H.**

بسم الله الرحمن الرحيم

اللهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِن شَجَرَةٍ مُّبَارَكَةٍ زَيْتُونِةٍ لَّا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نُورٌ عَلَى نُورٍ يَهْدِي اللهُ لِنُورِهِ مَن يَشَاء وَيَضْرِبُ اللهُ الْأَمْثَالَ لِلنَّاسِ وَاللهُ بِكُلِّ شَيْءٍ عَلِيمٌ(٣٥)

صدق الله العلي العظيم

سورة النور– اية ٣٥

Declaration

I hereby declare that this dissertation, submitted to the University of Babylon as fulfillment of requirements for the degree of doctor of Philosophy in Information Technology\ Software has not been submitted as an exercise for a similar degree at any other university. I also certify that the work described here is entirely my own.

Signature:

Name: Hafedh Ali Shabat Mizael

Date: \ \ 2021

Supervisor Certification

I certify that the dissertation entitled (**Improving the Independent Component Analysis Based on Metaheuristic Algorithms for Text Clustering**) was prepared under my supervision at the department of Software/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Doctor of Philosophy in Information Technology-Software.

Signature:

Supervisor Name: Prof. Dr. Nidaa Abdual Muhsin Abbas

Date: / / 2021

The Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled "**Improving the Independent Component Analysis Based on Metaheuristic Algorithms for Text Clustering**" for debate by the examination committee.

Signature:

Assist. Prof. Dr. Ahmed Saleem Abbas

Head of Software Department

Date: / / 2021

Certification of the Examination Committee

We, the undersigned, certify that (Hafedh Ali Shabat Mizael) candidate for the degree of Doctor of Philosophy in Information Technology-Software, has presented his dissertation of the following title (**Improving the Independent Component Analysis Based on Metaheuristic Algorithms for Text Clustering**) as it appears on the title page and front cover of the dissertation that the said dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: (21/10/2021).

Signature:
Name: Dr. Kadhim Mahdi Hashim
Title: Professor
Date: / / 2021
(**Chairman**)

Signature:
Name: Dr. Hussein K. Al-Khafaji
Title: Professor
Date: / / 2021
(**Member**)

Signature:
Name: Dr. Nidhal K. El-Abbadi
Title: Professor
Date: / / 2021
(**Member**)

Signature:
Name: Dr. Jane Jaleel Stephan
Title: Assistant Professor
Date: / / 2021
(**Member**)

Signature:
Name: Dr. Enas Hamood Al-Saadi
Title: Assistant Professor
Date: / / 2021
(**Member**)

Signature:
Name: Dr. Nidaa Abdual Muhsin Abbas
Title: Professor
Date: / / 2021
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Dr. Hussein Atiya Lafta
Title: Professor
Date: / / 2021
(Dean of Collage of Information Technology)

Dedication

I dedicate this work to my sweet and loving father and mother. I also dedicate the profit of my effort to the soul of my dear brother martyr may God dwell in his spacious gardens, I also dedicate profit of my effort to those who supported and encouraged me in my research, my wife, brothers, sisters, and friends.

Acknowledgement

First and foremost, I would like to thank my God, Allah Almighty, for giving me endless graces. My deep sense of gratitude to the beacon of science, to the master of creatures, to the greatest Prophet, Mohammed (Peace be upon Him and His Family).

I take this opportunity to express my sincere gratitude and greatest appreciation to my supervisor Prof. Dr. Nidaa A. Abbas for her continuous support for my Ph.D. Thanks for her the effort, follow-up, and supervision she provided for my dissertation during the research period.

Many thanks to everyone who helped me in the College of Information Technology, Department of Software, to complete my research work. Finally, many thanks to the University of Babylon for giving students the opportunity to obtain higher degrees in education for the progress and development of the country.

Abstract

The independent component analysis (ICA) is a widely used method for solving blind source separation problems where assumes that the sources are independent of each other and extracts them by maximizing their non-Gaussianity as the objective function. ICA comprises two components, the optimization method, and the objective function. Most popular ICA methods are used gradient function as an objective function. The disadvantage of these methods is that they are trappable at a local minimum.

Metaheuristic algorithms have a number of advantages over conventional heuristic techniques, including ease of implementation and a large potential for escaping local optimum solutions. Therefore two Metaheuristic optimization algorithms(Particle Swarm Optimization(PSO) and Glowworm Swarm Optimization (GSO)) can be proposed for address the local minimum drawback and improve the performance of the ICA method.

These proposed methods are implemented to propose a text clustering system. The documents can be understood as mixtures of latent concepts grouping the words. Therefore, Singular Value Decomposition (SVD) in Latent Semantic Analysis is employed for transforming words and documents into a new dimensional space that reflects semantic concepts. The largest principal components of SVD analysis were used as input to the FastICA, PSO-ICA and, GSO-ICA algorithms for extracting the Independent Components (ICs) which were used as clusters. To quantify the ability of the ICA to cluster documents,

the ICs are converted into “clusters probabilities” using the Argmax function.

For evaluating the performance of the proposed methods, typical assessments metrics were used involve (precision, recall, F-measure, and Overall Accuracy). Three main experiments were conducted, the first experiment was conducted on the medical dataset and the other experiments were conducted on the BBC news dataset. The results obtained using a medical dataset were 90.3% for overall accuracy with elapsed time=0:00:08.328817 using the GSO-ICA algorithm, 89.5% for overall accuracy with elapsed time= 0:00:07.222813 using the PSO-ICA algorithm, and 85.5% for overall accuracy with elapsed time=0:00:06.927412 using the FastICA algorithm. The results obtained using the BBC news dataset were 91.6% with time=0:00:25.880445 using the GSO-ICA, 90.8% with time=0:00:24.819643 using the PSO-ICA, and 87.2% with time= 0:00:22.713640 using the FastICA for a subset of texts of the BBC news dataset. The last experiment is conducted on the entire BBC news dataset, the results were 90.3% with time= 0:03:33.767175 using the GSO-ICA, 90.1% with time= 0:03:21.817554 using the PSO-ICA, and 84.9% with time= 0:02:51.923489 using the FastICA.

These results depict the outperformance of the proposed algorithms compared with the standard FastICA algorithm. In addition, the experimental results showed that the GSO-ICA superior on PSO-ICA due to ability to its split agents into subgroups and find multiple global solutions simultaneously.

Declaration Associated with this Thesis

Some of the works presented in this dissertation have been published as listed below.

1. The work entitled " Independent Component Analysis Based on Natural Gradient Algorithm for Text Mining" has been published in Information Technology to Enhance e-learning and application (IT-ELA) conference in 2020, IEEE Publishing. The classification of this publishing is related to the Scopus database.

2. The work entitled " Enhance the performance of independent component analysis for text classification by using particle swarm optimization" has been published in International Conference on Advanced Science and Engineering 2020 (ICOASE), IEEE Publishing. The classification of this publishing is related to the Scopus database.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

Abbreviation	Description
ACC	Accuracy
ANN	Artificial Neural Networks
ARI	Adjusted Rand Index
BIC	Bayes Information Criterion
BOC	Bag of Concepts
BSS	Blind Source Separation
DM	Data Mining
EM	Expectation-Maximization Method
FN	False Negatives
FP	False Positives
GloVe	Global Vectors for Word Representation
GNMF	Graph Regularized Non-negative Matrix Factorization
NMF	Non-negative Matrix Factorization
GSO	Glowworm Optimization Swarm
GWO	Grey Wolf Optimization
ICA	Independent Component Analysis
Ics	Independent Components
k-NN	k-Nearest Neighbor
LR	Logistic Regression
LSA	Latent Semantic Analysis

LSI	Latent Semantic Indexing
MED	MED Abstract
MHOAs	Metaheuristic Optimization Algorithms
ML	Maximum Likelihood
NB	Naive Bayes
NLP	Natural Language Processing
NMI	Normalized Mutual Information
P	Precision
PSO	Particle Swarm Optimization
R	Recall
SI	Swarm Intelligence
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TM	Text mining
TN	True Negatives
TP	True Positives
VSM	Vector Space Model

List of Symbols

Symbol	Description
S	Sources, Components
X	Observations (mixed signals)
A	Mixing Matrix
W	Separation Matrix
y	Separated Signals
E	Expectation
K	Number of the sources
P(x)	Probability of x
H(x)	Entropy of x
J(x)	Negentropy of x
MI	Mutual information
U	Column-orthonormal matrix
∑	Diagonal matrix
V	Column-orthonormal matrix
VT	Transpose matrix
log2	Logarithm of base 2
G	Logcosh function
g′	Derivative of g function.
Xc	Centering data
Xw	Whitening data

CHAPTER ONE

GENERAL INTRODUCTION

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Overview

The purpose of blind source separation (BSS) is to recover the underlying components from their mixtures where the mixing matrix and components distribution are unknown. To address this problem, independent component analysis (ICA) is the most often used technique for extracting those components under the assumption of statistically independence [1]. ICA was initially designed to solve the problem of blind source separation. Its goal is to recover separate source signals from linear mixes of these signals as accurately as possible [2]. The process of independent component analysis involves linearly transforming multivariate data (observations) in order to minimize dependencies between variables, resulting in so-called independent components (ICs). As a result, the objective of ICA is to identify both the separating matrix and sources (ICs), and the problem is not identifiable if more than one component has a Gaussian distribution [3].

ICA is currently a well-known concept that has effectively expanded across a diverse range of scientific fields and applications throughout the years [4]. It began with processing for signal processing applications involving communications, medical signal processing, speech signal processing, and so on [5-7]. Lately, it was discovered that the ICA algorithm is an effective technique for the problem of finding the latent structure within high-dimensional data, which is important of data mining. The observed data are thought to be the result of unknown

latent variables and their interactions [8]. The task of the ICA algorithm is to find these latent variables using just observed data.

Data mining (DM) is a process for extracting useful information from a huge amount of raw data. In past two decades, advances in data mining in several domains such as machine learning, artificial intelligence, and statistics have prompted academics to design and use new data mining strategies and approaches. Web mining, text mining, educational virtual platforms, and research publishing databases are examples of trends and areas where data mining methods may be used to assist humans in making decisions [9].

Text mining (TM) is a process for extracting meaningful and interesting knowledge from textual data sources [10]. Application areas like search engines, customer relationship management system, filter emails, product suggestion analysis, fraud detection, and social media analytics use text mining for opinion mining, feature extraction, sentiment, predictive, and trend analysis [11]. The important step of text mining perform is collecting data, where collecting unstructured data is done from a variety of sources in different file formats, including plain text (documents), web pages, and pdf files, among others.

There are a variety of text mining methods that may be used to analyze text data. Clustering is an unsupervised technique that uses clustering algorithms to organize text documents into groups. ICA is an unsupervised approach, which means there is no labeled training data to assist in sorting collections of texts into subsets. ICA can be applied to

analyze text data once it has been converted to a suitable numerical format. Using ICA for text analysis is based on the assumption that a document collection (also known as a corpus) is created via a mixture of several topics [12, 13]. In addition, according to [14, 15], in document analysis studies, a term by document matrix is considered as the linear mixtures of a set of independent sources.

Most conventional ICA algorithms used gradient descent methods. These gradient-based approaches employ gradient functions as objective functions for estimating independent components [16]. The quality of the gradients functions suffers from stucking in the local minima of the search space [17, 18]. To overwhelm the disadvantage, two meta-heuristic algorithms particle swarm optimization (PSO), and Glowworm optimization swarm(GSO), are employed in the ICA algorithm [19, 20]. In statistics, negative entropy (also known as negentropy) is the measurement of the distance to normality. To enhance the performance of the ICA method for text clustering model, as one of the applications of text mining, the current dissertation utilized Meta-heuristic methods which employed the negentropy function as a fitness function.

The conventional evaluations were employed to evaluate the suggested techniques' performance. Recall, precision, F-measure, and macro-average are common examples of these measures. True positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), are all represented in the confusion matrix [21].

1.2 Related works

In this section, the literature that introduced methods for text analytic was displayed in the following two subsections. The first one introduces a survey of using ICA for text clustering as one application of text mining. The second introduces a display of using Metaheuristic algorithms for text mining.

1.2.1 ICA for Text Clustering

Clustering is data mining approach that is very important, especially when dealing with huge amounts of data. When applied to text documents, it automatically places those documents that have similar themes together and separates those that have different themes. ICA algorithm has been used for text clustering in several previous studies.

Wang et al. (2006), proposed a strategy to cluster XML documents based on ICA. This strategy based on ICA (called ICAXC) which consists of three stages: vector representation, feature transformation with ICA and clustering with C-means method. Where ICA reduces the dimensionality of vector space and finds the latent variables. It can mine the projection axes that can be aligned with the data distribution and embody more information. Experimental results show that the method (C-means algorithm) using Independent Component Analysis outperformed the standard C-means clustering method. Where the error of standard C-means =18% whereas the error of ICAXC=145 [22].

Huang et al. (2009), introduced a method that used Wikipedia and semantic knowledge that contains for document clustering. The authors created a concept-based document-representation of Wikipedia articles called a bag of concepts (BOC) document representation. They applied latent semantic indexing (LSI) and ICA to Wikipedia articles to build a BOC model and use the identified latent concept structures for clustering. Where they used the semantic relatedness between concepts to calculate the similarity between documents. This approach tested the concept-based representation and the similarity measure on two standard text document datasets: OHSUMed and Reuters. The results proved the effectiveness of proposed BOC model. Where the performance of LSI and ICA on BOC model on Reuters dataset. The F-measure of BOC+LSI was 0.195 and BOC+ICA was 0.201[23].

Pu, Q., and D. He. (2009), Proposed a clustering approach using ICA algorithm and relevance model based on semantic clustering. After construct the term-document matrix they used Bayes Information Criterion (BIC) to get the K values that reflect the latent topics in a collection of documents. The ICA calculates the separating matrix using the maximum likelihood principle, then used it to find sources and after that applied softmax function which described the degree that document belonged to latent topic. Therefore can be used the ICs as clusters of documents. The information of this clusters employ into process of relevance model [24].

Onoda et al. (2012), proposed the independent component analysis-based seeding method, and compare the proposed method with

three others clustering methods which are (k-means, KKZ method, and k-means++). K-means clustering method sometimes generates bad clusters because it is based on the choice of the initial center which is chosen uniformly from random data. This choice may be the bad center, therefore, lead to a bad clustering process. In the suggested technique, ICA will extract ICs and use each IC as an initial seed in order to generate the global optimum case of clustering. Two types of datasets(ODP web corpus and UCI dataset) were used to assess the suggested method's performance. Normalized mutual information (NMI) used as a metric to evaluate the qualities of clustering outputs of different methods. The results to ODP web corpus using NMI metric for KKZ was 0.531, 0.514 for k-means, 0.525 for k-means++, whereas was 0.638 for ICA. Experiments indicate that the suggested approach outperforms other methods for clustering datasets [25].

Gultepe et al. (2018), used some standard clustering algorithms with six benchmark datasets, the (REUTERS-10K) text document corpus is one of them. Clustering was done using methods including: K-means, spectral clustering, Graph Regularized Non-negative Matrix Factorization (GNMF), and K-means with PCA. Two ways are proposed to improve the performance of these algorithms. The first one is the blind source separation using ICA was employed with each clustering method. The second improves features for constructing the distance matrix of clustering algorithms based on graph by extraction the features using deep learning-inspired feature learning approaches. After conducting the experiments and using assessing metrics, When PCA

dimension reduction followed by ICA BSS is applied directly applied to the tf-idf matrix, K-means clustering provides the top clustering performance (NMI=0.46, ACC=0.714). Without ICA BSS, PCA dimension reduction provides an accuracy of (NMI=0.446, ACC= 0.656). The next best clustering performance was provided by NMF without (NMI=0.318, ACC=0.546) and with ICA BSS (NMI=0.428, ACC=0.638) [26].

Ghazdali et al. (2021), presented ICA algorithm-based approach for document clustering and topic identification. They used data from Wikipedia articles in their analysis; each article is considered as one document, and the corpus is a collection of these documents. Three clusters of data about machine learning, video games, and black holes were recognized [27].

1.2.2 Metaheuristic Algorithms for Text Clustering

As a result of their ability to find better solutions for clustering analysis problems, nature-inspired optimization algorithms have a lot of interest. Text clustering has been accomplished via the use of Metaheuristic algorithms in a number of previous researches.

Hasanzadeh and Rokny (2012), proposed a method of PSO algorithm based on the latent semantic indexing model. In addition, they used adaptive inertia weight in the PSO algorithm, which may result in effective exploration and exploitation of the search space as well as fast convergence. To illustrate that the suggested algorithm is effective, the experiments were conducted on the two benchmark datasets (Reuters

and Hamshahri). Analyzes indicated that the combination of PSO with LSI provides produces better clustering accuracy and efficiency, and is faster than the PSO+Kmeans algorithm [28].

Karol and Veenu, (2013), presented two methods for effective text documents clustering by hybridizing the conventional partitioning clustering methods K-Means, and Fuzzy-C Means with PSO. The suggested methods were tested on two datasets (Reuters-21578, and 20NewsGroup). The performance of these hybrid algorithms have been evaluated in comparison to the performance of standard methods K-Means and Fuzzy-CMeans(FCM), and the results have been encouraging. Where results in Reuters-21578 dataset showed that KPSO and FCPSO give approximately 37% better results than KMeans and FCM for Entropy, approximately 17% better result than KMeans and approximately 18% better values than FCM for F-Measure. Also FCPSO and K-Means provide approximately 14.66% better values for F-Measure and 16.5% better values for Entropy than K-Means and FCM algorithm as results in 20NewsGroup dataset. These results shows that proposed methods FCPSO and KPSO provide superior outcomes comparing with standard algorithms [29].

Abualigah et al. (2018), proposed a method for feature selection based on the PSO algorithm. This technique known as FSPSOTC for solving the features selection problem by generating a new subset of informative text features, thereby improving the effectiveness of the text clustering method and reducing computational time. When compared it

to other well-known algorithms such as the evolutionary algorithm and harmony search method, the proposed approach outperformed them[30].

Janani et al. (2019), developed a text clustering method known as SCPSO which combines Spectral Clustering and PSO to enhance text clustering. The spectral clustering with swarm optimization produce method to deal with the large volume of text documents, through a way in which randomization is carried performed with respect to the initial population by taking into account both local (finding the optimal solution for a specific region of the search space) and global(finding the optimal solution on problems that contain local optima) optimization functions. The proposed SCPSO was compared against other methods such as Expectation-Maximization Method (EM), Spherical K-means, and standard PSO algorithm, using benchmark database. The results demonstrated that the proposed SCPSO method outperforms others clustering approaches in terms of clustering accuracy. The SCPSO algorithm increases by 6% in terms of accuracy, 8% increase in terms of NMI and 9% increase in term of ARI when compared to PSO clustering algorithm in Reuters dataset. In 20Newsgroup dataset, there is 7% increase in accuracy, 9% increase in terms of NMI and 9% increase in ARI when compared to the PSO clustering algorithm. The 6% of increase in accuracy, 8% increase in NMI and in terms of ARI, there is a 9% of increased results in SCPSO. Hence the proposed algorithm yields better performance in terms of clustering the documents [31].

Selvaraj et. al. (2021), employed several algorithms such as the PSO, Bat, grey wolf optimization (GWO), and K-means techniques for

performing text document clustering process. In the same time, conducted a comparative study among them using six datasets of different sizes. Based on the results of the experiments, the PSO and GWO algorithms outperform K-means, and the PSO outperforms the other of Metaheuristic algorithms in terms of identifying the optimum solution for document clustering [32].

1.3 Problem Statement

Many of the ICA algorithms have the basic form optimizing an objective function based on the gradient-descent. Gradient descent will lead to the closest local minimum instead of a global minimum. That is will be getting the local solution not the best solution. That is means will not be getting the best solution (global solution).

Metaheuristic algorithms offer many benefits over other heuristic techniques, including the fact that they are simple to implement and have a high capacity to escape local optimum solutions. Therefore to alleviate this drawback, improved ICA algorithms based on Metaheuristic algorithms are presented. The strategy can enhance the ICA algorithm capability by using the PSO and GSO algorithms to determine the optimum global solution and avoid traps at the local optimum.

The essential idea is that ICA based on Metaheuristic methods represented by PSO and GSO algorithms which used negative entropy as objective function to replace the gradient function in the ICA algorithm

to find the best separating matrix. Then, the independent components can be extracted.

The proposed Metaheuristic based ICA algorithms provided superior separation outcomes than the standard FastICA algorithm that was used as a baseline for comparison, according to experimental findings using textual data. In addition, experimental results showed that the proposed PSO–ICA and GSO-ICA approaches are effective ICA algorithms.

1.4 Motivations

1. Text data represent a good example of unstructured content, which can be generated as one of the simplest types of data. It is easily interpreted and understood by humans, but is considerably more difficult to comprehend by machines. It is therefore essential to employ effective algorithms to process these data efficiently.
2. Most machine learning approach do some kind of similarity measure to distinguish between two vectors, but due to the high dimensionality of the feature vector in text clustering, these similarity measures lose their discriminative strength. Thus, employ ICA with LSI to address this drawback.
3. The similarity between two text documents is measured as the cosine coefficient between their term frequency vectors. However, a major drawback of the words-based approach is its inability of handling the polysemy and synonymy phenomenon of the natural language.

Polysemy refers to the issue of more than one interpretation of words, while Synonymy is used to explain the fact that multiple words share the same meanings. therefore, used ICA with LSI to produce text clustering based on the concept to overcome this phenomena.

4. ICA algorithm is based on neural network concept. Thus, it suffering a local minimum problem. PSO and GSO Metaheuristic algorithms are proposed to overcome this dilemma and enhanced performance of ICA algorithm.

1.5 Aims of Dissertation

1. Propose and develop a text clustering system based on statistical method by using independent component analysis unsupervised machine learning.
2. Enhancing the performance ICA algorithm by using two Metaheuristic optimization algorithms: PSO and GSO.
3. Using negentropy function as an objective function in ICA method to propose text clustering system.

1.6 Contributions

1. The performance of the ICA algorithm were improved by using Metaheuristic algorithms such as particle swarm optimization (PSO) and Glowworm swarm optimization (GSO).
2. Employ the *negentropy* function as objective function of PSO and GSO algorithms to enhance the measure of ICA algorithm performance in the text mining field.

3. Produce text clustering model based on the concepts to address the ambiguity which it suffers the text clustering methods which based on single-word similarity measures due to polysemy and synonymy.

1.7 Structure of Dissertation

After this chapter, the rest of this dissertation is:

- **Chapter Two:** This chapter provides the introduction of text mining and text clustering, an overview of ICA as a statistical method. With mathematics, the focus is on understanding the fundamental mathematical concepts. We will give brief coverage of Metaheuristic methods especially to those algorithms that are proposed to improving the ICA method. Also, this chapter presents the evaluation metrics used in the proposed system.
- **Chapter Three:** focuses on the proposed system design in details and the algorithms used in the system.
- **Chapter Four:** represents the results of the system which are discussed in details.
- **Chapter Five:** discusses the conclusions and suggestions for future work.

CHAPTER TWO

THEORETICAL FOUNDATION

CHAPTER TWO

THEORETICAL FOUNDATION

2.1 Introduction

The intention of this chapter is to provide an introduction to text mining. Text clustering will be discussed as one of the tasks associated with text mining. Independent component analysis algorithm will be mentioned as the main approach used in this dissertation. Also, Metaheuristic algorithms will be covering which are used to improve the performance of the ICA algorithm. Lastly, presented the evaluation metrics which will use to evaluate the performance of the proposed model.

2.2 Text Mining

Text mining (TM) is described as the manner of extracting the implicit knowledge from textual data [33]. It is necessary to differentiate between implicit knowledge produced by text mining and information obtained from the storage since the implicit knowledge produced by text mining does not exist in the supplied storage. TM is primarily concerned with the tasks of text clustering, text categorization, and text association. TM is a special kind of data mining that is distinct from the others, therefore, the purpose of this section is to render a general understanding of text mining.

Text is defined as unstructured data that is made up of strings which are called words [34]. Even though a collection of words is considered text in the widest sense, the meanings of the individual

strings and their arrangement by rules, known as grammars, are required in order for the text to be created.

The key traditional tasks in text mining are classification, regression, clustering, and association [35]. Due to ICA is unsupervised machine learning, therefore, we will describe clustering as an unsupervised application in this dissertation, and proposed a text clustering system based on the ICA algorithm.

2.3 Text Clustering

TM refers to the special kind of data mining where texts is presented as the source. In light of the increasing availability of electronic documents from a diverse variety of sources, text mining researches have become more important. Which include the use of unstructured such as text files and semi-structured data HTML files.

Text Mining's key aim is to help users to retrieve information from textual resources, which can perform an operations such as extraction, classification, clustering, and summarization. Natural language processing (NLP), data mining, and machine learning methods work together to categorize and discover patterns from different kinds of texts automatically [36].

Text clustering is the process of segmenting a collection of texts into partitions where the texts in the same group (cluster) are more similar to each other than texts in other clusters [37]. It is usually

accomplished via the use of a variety of text manipulation techniques and specialized algorithms in order to identify patterns and trends. Figure 2.1 provides a form of text clustering.

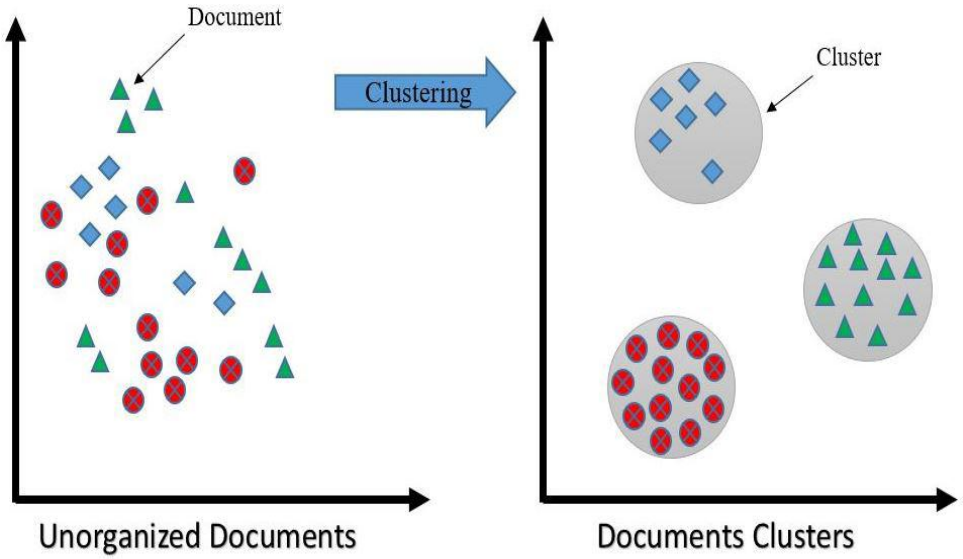


Figure 2.1: Text Clustering [38]

In the text domain, the clustering problem can be extremely useful. Where the objects to be clustered may be of various granularities such as documents, paragraphs, phrases, or words. Text clustering is a useful technique that aims to organize large document collections into smaller meaningful and manageable groups, and it plays an important role in document organization, information retrieval, and topic extraction [39]. Automatic text clustering consists of text representation, feature extraction or transformation, employment of text mining

algorithm, and finally an evaluation of the applied text mining algorithm.

2.4 Text Clustering Process

The main stages of text clustering include: text representation, feature extraction, catching a Vector Space Model (VSM), implementing a text mining algorithm, and finally evaluating the text clustering as showing in Figure 2.2. These stages will be presenting in the next sections in details:

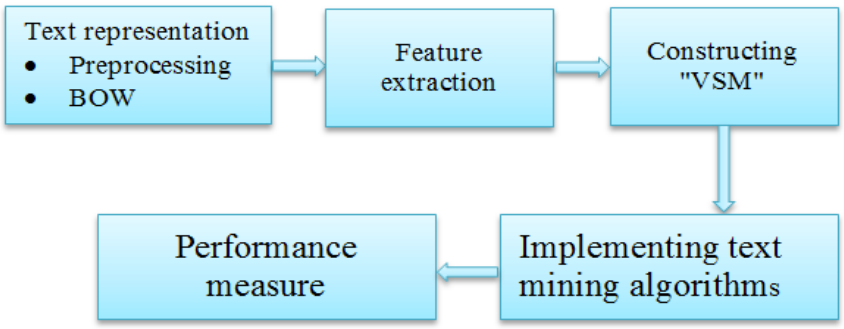


Figure 2.2: Main Stages of Text Clustering [40].

2.4.1 Text Representation

The representation of a document is the process of presenting a document in a manner that is acceptable for the data mining system. After documents collection, the initial step of the clustering process is indexing. The conversion of a text or collection of texts into a list of Words is known as text indexing [41]. Since a document or text itself is

basically offered as unstructured forms, it is practically difficult to really process its raw form using a mathematical model. In other words, the indexation of text involves the segmentation of a text composed of phrases into included words and the outcome of indexing the text is a list of words, in order to handle the text easily manner. Figure 2.3, demonstrates the three fundamental stages of text indexing. The first stage is the method of splitting the text based on white spaces or marks of punctuation into words (tokens). The second stage is the method of converting each word using morphology rules into its own root, this process is called stemming. The last step is to eliminate the stopword that is to erase the morphological words such as prepositions, articles and, conjunctions.

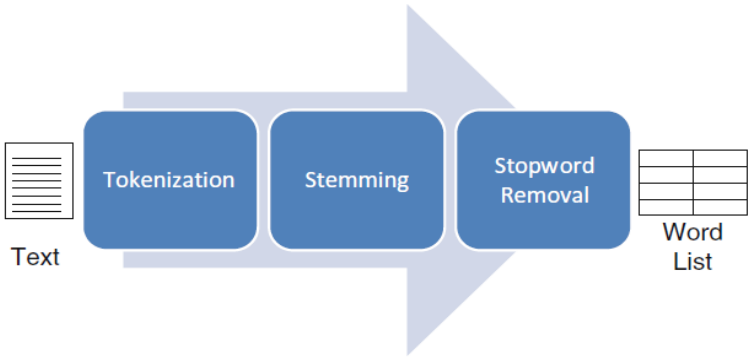


Figure 2.3: Basic Stages of Text Indexing [42].

2.4.1.1 Tokenization

The process of segmenting a text or document into tokens using white space or based on punctuation marks is known as tokenization

process [43]. As long as texts in a natural language have been written, the scope of this dissertation is restricted to just text since it relies on two datasets written in English. As a result, the work will focus on how to tokenize English-language texts.

The tokenization functional performance is shown in Figure 2.4. The text is given as an input and the tokens list is created as a result, where the text is divided based on white space or using punctuation marks into words (tokens). When subsequently processed, the words with particular characters such as “82%” or digital values are discarded by using regular expression, and the tokens are replaced to lower case characters. These tokens are used as input for the next step of text indexing, which includes the stemming and elimination stopword.

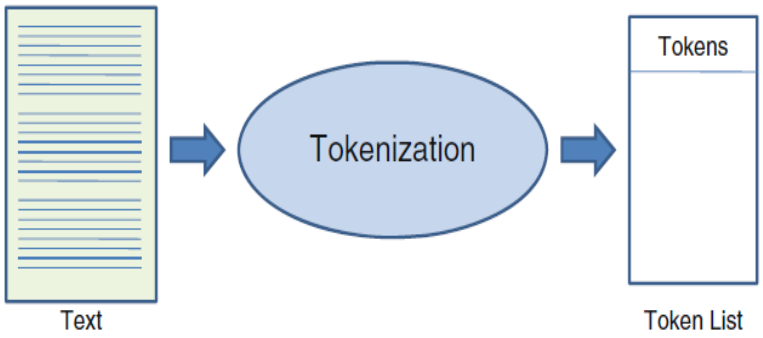


Figure 2.4: Tokenization Process [42].

2.4.1.2 Stemming

In the area of natural language processing, stemming is widely utilized as a second preprocessing step [41, 44]. Stemming indicates a

method of converting each word that is produced from the preceding stage (Tokenization) into the shape of its own root. Porter algorithm is one method used to find the stem of the words. Table 2.1 displays an example of the result stemming process.

Table 2.1: Example of Converted Terms to Their Root [42].

Varied form	Root form
complexity	complex
assigned	assign
assigning	assign
categorizes	categorize
categorizing	categorize

2.4.1.3 Stopwords Removal

Stopwords removal indicates the process of eliminating stopwords from the list of tokens (words) or stemmed words [41]. Stopwords are grammatical words that are not related to the content of the text and thus must be eliminated in order to increase efficiency. The list of stopwords is loaded from a stored file. The words in the text that are already displayed in the list stopword are deleted. It is possible to swap the stemming and stopwords removal processes, such that stopwords are removed first and then the tokens are stemmed.

The term "stopword" refers to a word that serves solely grammatically and has little or no significance in the context of the text

contents, such as Prepositions, such as “on”, “to” ,“in”; and so on. Conjunctions such as “however”, “and”, “or", and “but". In addition, stopwords that are more often used include the definite article "the," as well as the infinite articles "a" and "an."

Stopwords appear dominantly in texts in the corpus; eliminating them improves text processing performance significantly. The remaining words are important and meaningful. Finally, a list of words derived from text indexing is given as inputs to the text encoding process, which will be used to encode the text.

2.4.2 Feature Extraction

Texts and documents are unstructured data. So, when using mathematical modeling to cluster unstructured texts, they must be transformed these texts into a structured feature space. First, the data must be cleaned to remove any unnecessary letters and words. After the data has been cleaned, conventional feature extraction techniques can be applied to extract features from the data**.**

The common techniques of feature extractions are Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF) [45], Word2Vec [46], and Global Vectors for Word Representation (GloVe) [47]. These approaches categorize as either word embedding or weighted word approaches.

The common fundamental form of extraction of weighted word features is TF, where each word is converted to a number (value) corresponding to the number of occurrences of that word in the entire text. Techniques that utilize term frequency as a Boolean or as a logarithmically, scaled weighting to extend the findings of TF are most often seen. In all weight words techniques, each text is converted into a vector (has length the same length of the text) containing the frequency of the word in that text [48].

2.4.3 Constructing a Vector Space Model (Text Encoding)

The process of mapping a text file into a structured form is referred to as the encoding of the text. The output of the preprocessing stage of text which was covered in the previous section will become the input to the process of encoding text. The extracted words from the stage of text indexing are selected as features, and the numerical values are assigned to them when encoded each text. As a result, the text will be represented by the numerical vector which is generated from this process. Figure 2.5 depicts the converting text process into a numerical vector.

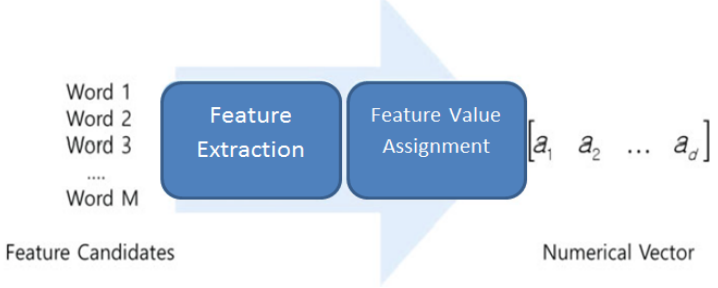


Figure 2.5: The Procedure for Converting Text into A numeric Vector[42].

In the text representation process, the vector space model (VSM) is widely employed [49]. Documents are converted into sparse numeric vectors using this approach. Each term in the VSM is presented as a separate variable with varying importance of numerical weight [50]. A vector can be used to describe each text. The terms in document j are $d_j = [w_{1j}\ w_{2j}\ \ldots..\ w_{mj}]$, where m is the number of terms in the list of terms. When all of the documents are combined into one matrix, the words and document represent the rows and columns, respectively. Equation (2.1) expresses the term-document matrix form as follows:

$$X= \begin{array}{c} \\ T_1 \\ T_2 \\ . \\ . \\ T_M \end{array} \begin{array}{c} \begin{array}{ccccc} D_1 & D_2 & D_3 & ..... & D_N \end{array} \\ \begin{bmatrix} w_{11} & w_{12} & w_{13} & .... & w_{1N} \\ w_{21} & w_{22} & w_{23} & .... & w_{2N} \\ . & & & & . \\ . & & & & . \\ w_{m1} & w_{m2} & w_{m3} & .... & w_{MN} \end{bmatrix} \end{array} \quad .....(2.1)$$

Where T is represent the terms, D is represent the text, M represent the number of words, and N represent the number of texts. The occurrences of terms corresponding to features in the given document are assigned as values of features in different schema, in this dissertation, the weighting method used to calculate the weight of each word in a document is expressed in Equation (2.2):

$$W_{ij} = \frac{f_{ij}}{\sum_{i=1}^{M} f_{ij}} \quad \ldots\ldots \quad (2.2)$$

Where f_{ij} represents the frequency of the word i which appears in the text j. This schema is used to determine the significance of words in the collection of texts (dataset).

2.5 Text Clustering and Machine Learning Algorithms

The central stage in the text mining framework is to select the appropriate algorithms. These algorithms can be classified into two main kinds of machine learning algorithms such as supervised (text classification), and unsupervised (text clustering) [51].

Supervised machine learning techniques are depended on the assumption that the structure of a text dataset has already been determined (annotated). They need a training set of labelled documents, as well as the declaration of a function that maps documents to the predefined class's labels. The most popular supervised algorithms are: The logistic regression (LR) is one of the easiest classification algorithms to be discussed in most areas of data mining [52], k-nearest neighbor (k-NN) is a successful technique that has been studied and used as a classification task [53], Naive Bayes(NB) is considered the oldest information retrieval method as a viable application [54], and support vector machines (SVM) is another common method which employs a discriminative classifier for document classification [55, 56].

Unsupervised machine learning algorithms are also referred to as clustering algorithms and are generally machine learning with no predefined labels used. In unsupervised algorithms, the input data is an

unlabeled collection of texts. The aim is to cluster documents together without the need for previous knowledge, such that documents inside a cluster are more similar to one another than documents between clusters. Traditional clustering methods can be classified into two main groups as partitioned and hierarchical algorithms. The common successful unsupervised algorithms are: K-Means is probably the most well-known clustering algorithm [57].

ICA is unsupervised machine learning was widely employed for blind source separation. Also, it has been employed in several a variety of applications [58], for example, latent variable decompositions, weather data mining, extracting latent signals from pictures of satellites, and text data analysis [8]. In this dissertation, the ICA algorithm will be introduced as a method to propose a system for text clustering.

2.6 Blind Source Separation

In the real world, it is possible to receive mixed signals from everywhere. Therefore, it is often challenging to get accurate information from environmental sources. For instance, sounds are accompanied by distortion caused by the echo of the room, when several sources are active at the same time. That is, the hearing conditions for voice signals have degraded significantly. Another example, according to the viewpoint of image processing, the observed image becomes blurred with noise or mixed with the other image as a result of the

reflection of light. As a result, it becomes more difficult to detect and recognize the target object[59].

According to the assumption that a text dataset (collection of texts) is formed via a mixture of several topics [13]. Also, after converting this dataset into the numerical structure (term-document matrix) this structure is considered as a linear mixture of a set of independent sources. Each source is considered as a signal and each value in the signal is reflected as a document [14, 15].

Blind source separation (BSS) is a method for extracting a set of source signals from a set of mixed signals, where the information about the source signals or the mixing process is not known or only has a little amount. BSS is addressing the dilemma of signal recovery from the mixed-signal or the set of mixed signals. This is a multidisciplinary scientific field. Both signal processing and machine learning are two professional domains that have been broadly explored to deal with various challenges in BSS [59]. A famous instance of a source recovering problem is the cocktail party dilemma, in which a listener is trying to follow one of several conversations taking place in a room during attending a cocktail party [59]. As shown in Figure 2.6, three speakers (s1, s2, s3) are talking at the same time.

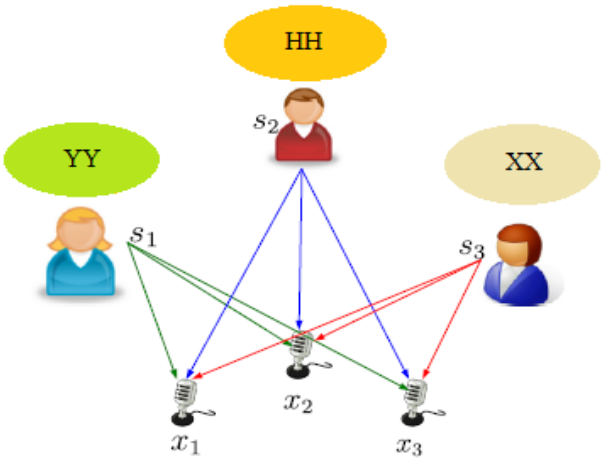


Figure 2.6: Cocktail Party Problem with Three Speakers and Three Microphones[59].

Three microphones (x_1, x_2, x_3) are placed close as sensors to collect signals of speech, that are mixed differently based on position, angle, and microphone channels properties. A linear mixing system is constructed as:

$$x_1 = a_{11}s_1 + a_{12}s_2 + a_{13}s_3$$
$$x_2 = a_{21}s_1 + a_{22}s_2 + a_{23}s_3 \quad \ldots\ldots\ldots \quad (2.3)$$
$$x_3 = a_{31}s_1 + a_{32}s_2 + a_{33}s_3$$

This 3 × 3 mixing system can be represented as X = AS in vector and matrix form, where $\mathbf{X} = [x_1\ x_2\ x_3]^T$, $\mathbf{S} = [s_1\ s_2\ s_3]^T$ and $\mathbf{A} = [a_{ij}] \in R^{3\times3}$. This system involves A as a constant mixing matrix, and the noise impact is not taken into consideration. It's also known as the noiseless and instantaneous mixing system. The sources are identified by estimation the demixing matrix W which recovers the original sources S

correctly, from the mixed observations X. Figure 2.7 illustrates a general linear mixing and demixing system.

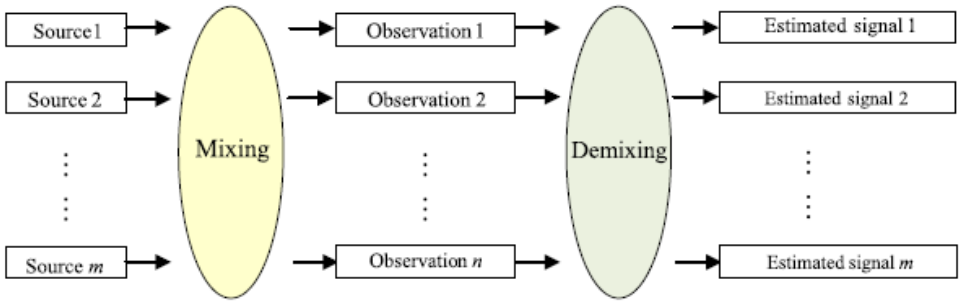


Figure 2.7: A general Linear Mixing and Demixing System [59].

2.6.1 Independent Component Analysis (ICA)

Historically, traditional BSS was developed by identifying a set of independent source signals from the mixed signals which were obtained from a number of sensors. The demixed signals were estimated by maximizing the measure of independence expressed in different ways. ICA was suggested in [3] to handle the cocktail party problem, in which mixed signals were collected using several channels or microphones. Assume you are at a dinner party where there is a lot of discussion and noise, and someone speaks to you. In this situation, the sound of the talker is very sensitive to your ear. This is the cocktail party problem that ICA aims to solve it. In order to deconstruct the mixed signals into individual sources, an estimation of a demixing matrix was performed, with the measure of the independence of the resulting being maximized.

The ICA unsupervised learning algorithm attempts to separate the observation vectors and find the salient features or mixture sources of the mixture. The aim of ICA is to extract independent components for individual sources. Because the resulting components are uncorrelated and independent (two random variables are uncorrelated if the expected value of their joint distribution is equal to the product of the expected values of their respective marginal distributions, whereas the independence means if the value of their joint distribution is equal to the product of the values of their marginal distributions), therefore, it is not possible to assume a Gaussian distribution in the ICA algorithm. As a consequence, there are some assumptions in ICA:

• The sources are statistically independent.

• Each independent component has a non-Gaussian distribution.

ICA discovers a set of hidden components that are mutually independent or non-Gaussian. The non-Gaussianity is mainly used as a measure of the degree of independence. Based on an information-theoretic principle, the independence or non-Gaussianity can be measured using mutual information (MI) and higher order statistics depend on kurtosis [60]. The typical ICA model can be describes in Equation (2.4):

$$X = AS \ldots\ldots\ldots (2.4)$$

Where X represents the linear mixture, A is the mixing matrix with unknown mixing coefficients, and S indicates the source signal. In Equation (2.5), the ICA estimates S as follows:

$$Y = W X \ldots\ldots. (2.5)$$

Where Y denotes the sources estimated (S) that are statistically independent and W indicates the demixing matrix. The goal, using a set of mixed signals X, to estimate the demixing matrix according to an objective function D(X,W), with the goal of obtaining recovered signals that are as close as to the original source signals.

2.6.2 Procedure of ICA Learning

ICA considers a linear representation of non-Gaussian data to be statistically independent or as independent as possible. ICA is commonly used for BSS, signal detection, text mining, and feature extraction [3]. The objective of ICA algorithms is to find the demixing matrix which provides the separation of mixture variables.

The demixing matrix W is estimated by minimizing or maximizing an objective function. In addition to kurtosis, there are many objective functions based on negentropy, entropy, mutual information, and likelihood function were employed for finding ICA solutions to the demixing matrix [60]. Figure 2.8 shows a procedure of standard learning of ICA for finding a demixing matrix W.

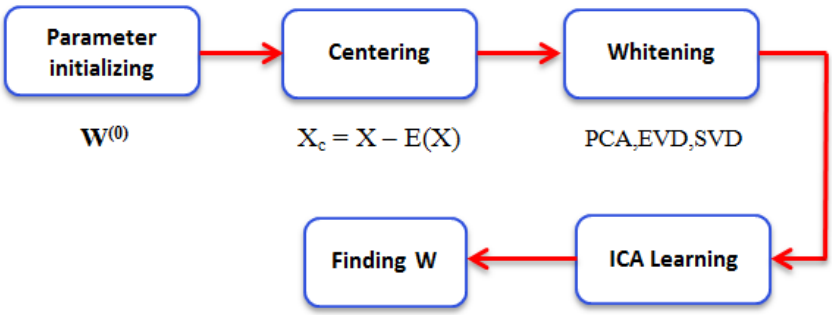


Figure 2.8: ICA Learning Process for Finding A demixing Matrix [59].

Firstly, start with an initial parameter $W^{(0)}$, then conduct preprocessing stage of a data, preprocessing consists of centering and whitening . Where each original sample in X is preprocessed by a mean removal operation, this process called centering of data by the Equation (2.6):

$$X_C = X - E(X) \ldots\ldots\ldots (2.6)$$

Where E(X) is expectation value(mean) of X. Then a whitening step that is conduct by one of the following technique (PCA, EVD, SVD). Centering and whitening processes are necessary to speed up the ICA algorithm. After that, running the ICA algorithm based on certain objective function, when the objective function satisfies a predefined condition, the learning procedure is terminated. Lastly, the demixing matrix was estimated and will be used to find the demixed signals from observed data by Equation (2.5).

2.6.3 Whitening

The next preprocessing stage after the centering step is the so-called whitening process. Also, it is called the sphering process [61]. Whitening transformation will produce uncorrelated variables which mean removes the second order dependences between the observed data. The goal of this process is to whiten the data by transforming vectors into other uncorrelated vectors and then rescaling each vector to have a unit variance. From this process, uncorrelated mixed data is obtained, as well as the observed data's unit variance [58]. In this dissertation, the observed data (text datasets) is represented by the vector space model (VSM). Latent semantic indexing (LSI) is the PCA of the VSM [62]. The singular value decomposition (SVD) is the well-known technique that will be used for this transformation, by using the SVD the term-document matrix is decomposed into singular values and singular vectors.

2.6.3.1 Latent Semantic Indexing (LSI)

LSI, also known as Latent Semantic Analysis (LSA), is a technique of analyzing a set of documents to discover statistical co-occurrences of words that appear together, which then give insights into the themes of those words and documents. Two problems that LSI sets out to solve are the points of synonymy and polysemy. Synonymy refers to the words that may be used to describe the same thing, while the word

with many meanings is referred to as polysemy, for example, the word jaguar can mean an animal, automobile, or an American football team.

LSI is an automated indexing technique that transforms words and documents into a low-dimensional space that reflects semantic concepts. It analyzes text at the conceptual level by projecting documents into a semantic space, to address the difficulties of using only term-based analysis [62]. The main objective of LSI is to analyze relationships between a collection of documents and the terms they contain by generating a set of concepts associated with the documents and terms. LSI is the PCA of the term-document matrix; the well-known technique to conduct this analysis is singular value decomposition (SVD) [8]. The idea is that the SVD defines a small number of "concepts" that connect the rows and columns of the matrix. Suppose X is the matrix with $m \times n$, the analysis by SVD can be shown in Figure 2.9:

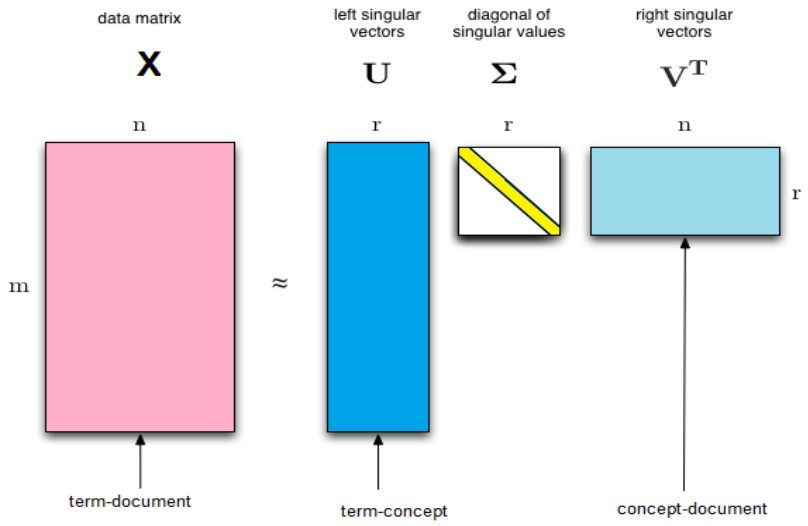


Figure 2.9: The Form of Singular Value Decomposition [63].

- U is an m× r column-orthonormal matrix; that is, each of its columns is a unit vector and the dot product of any two columns is 0.
- V is an n × r column-orthonormal matrix. Note that we always use V in its transposed form, so it is the rows of V^T that are orthonormal.
- ∑ is a diagonal matrix; that is, all elements not on the main diagonal are 0. The elements of ∑ are called the singular values of X.

In this dissertation, SVD is the effective technique used to decompose the term-document matrix (X), as shown in Equation (2.7) [63].

$$X = U\Sigma V^T \text{..... (2.7)}$$

Where U is a real number matrix of $m \times r$. Each column can be viewed as a concept which may be represent category or subject, etc. The particular combination of terms from the input with the weight of each term in the concept is indicated by the real number. Σ is the diagonal r × r matrix. V^T matrix is the new document representation of one text per row, each represented in terms of the concept specified in U instead of words occurring in the document. The matrix D describe in Equation (2.8).

$$D = \Sigma V^T \text{ …… (2.8)}$$

Where merges the concept weights and new representation of the document to indicate the scope in which the document obtains the concept.

At this stage, obtained the centering and whitening term-document matrix which represents the input data of the proposed system. The methods discussed in this dissertation estimates the structure in the data by finding latent components whose interactions might have generated the data.

2.7 ICA Objective Function

The objective functions of the ICA algorithm mainly includes the measurement of maximization non-Gaussianity (kurtosis and negentropy), minimization of mutual information, and maximum likelihood estimation. In ICA, the aim is to find the demixing matrix (W) and then projecting the whitened data onto that matrix for extracting independent signals. Three main approaches can be used to estimate this matrix. The first approach is dependent on non-Gaussianity, which can be measured using metrics like negentropy and kurtosis, and the goal of this approach is to obtain independent components that maximize non-Gaussianity [64]. Mutual information minimization is used in the second method to accomplish the ICA goal [65]. In the third approach, independent components can be estimated by using maximum likelihood (ML) estimation [66]. Simply, all approaches search for demixing matrix. Then the whitened data are projecting onto this demixing matrix lead to obtain independent components, the demixing matrix is approximated numerically through the optimization procedure.

To achieve the best approximation of the sources, all methods based on an optimization method and an objective function used with that method. In this dissertation, the Negentropy function is employed as the objective function in the proposed ICA based on metaheuristic methods.

2.8 Measures of Non-Gaussianity in the ICA

According to central limit theorem the distribution of a sum of independent signals with arbitrary distributions tends toward a Gaussian distribution under certain conditions. The distribution of the sum of two independent signals is typically closer to Gaussian than the distribution of the two original signals. As a result, a Gaussian signal is a linear combination of several separate signals [67]. This illustrates that the separation of independent signals from their mixtures can be achieved by making the linear signal transformation as non-Gaussian as possible. Non-Gaussianity is an important and essential principle in ICA estimation, in order to use non-Gaussianity in ICs estimation, there needs to be quantitative measure of non- Gaussianity of a signal. Searching for independent components can be achieved by maximizing the non-Gaussianity of extracted signals [68]. Kurtosis and negative entropy are two measures of non-Gaussianity that are used to determine the degree of non-Gaussianity. The next subsection will go through this in more detail.

2.8.1 Kurtosis

The Kurtosis can be used as a non-Gaussian measurement and the extracted signal can be obtained by finding the demixing vector, which maximizes the kurtosis of a signal extracted [69]. In other words, the source signals can be extracted by finding the orientation of the weight vectors which maximize the kurtosis.

The kurtosis of data which has been preprocessed to have unit variance is equal to the fourth moment. The Equation (2.9) [58] defines the Kurtosis of a signal (s), denoted by the symbol kurt (s):

$$Kurt(s) = E(s^4) - 3(E(s^2))^2 \ldots\ldots\ldots (2.9)$$

Where $E(s^2)$ is the variance , $E(s^4)$ is the fourth order moment. Basically, kurtosis is defined using a higher order "cumulant", which simplifies the formulation since it is dependent on the presumption that the signal has a zero mean. Furthermore, we may assume that signal(s) has been normalized to have a variance of one: $E(s^2)=1$. This will make things much simpler. Therefore, Equation (2.9) may be further simplified to provide the following Equation (2.10) [٥٨].

$$Kurt(s) = E(s^4) - 3 \ldots\ldots\ldots (2.10)$$

For a Gaussian signal the $E(s^4) = 3(E(s^2))^2$, and therefore its kurtosis is zero. For most non-Gaussian signals the kurtosis is nonzero. Kurtosis may be negative or positive. The kurtosis of super Gaussian random variables is positive, whereas the kurtosis of sub Gaussian

random variables is negative. The absolute value of kurtosis or the square of kurtosis is using a measure of non-Gaussianity [5].

2.8.2 Negative Entropy(negentropy)

Negative entropy, also known as negentropy, is a Gaussianity measurement. It depends on the amount of the theoretical information of the entropy. The entropy of a variable is a measure of its randomness. The entropy of a discrete variable is given by Equation (2.11) [6].

$$H(X) = -\sum P(X)\log_2 P(X) \quad \ldots\ldots(2.11)$$

Where H indicate for the entropy of the observation variables (original signals) as well as an estimated signals from the original variables (sources signals), P for the probability of X, and X represents the possible values of X. The Gaussian distribution variables have greater entropy than other variables.

The Negentropy concept defined to measure the Gaussianity of the components, as given in the Equation (2.12) [6, 58].

$$J(X) = H(X_{Gaussian}) - H(X) \quad \ldots\ldots(2.12)$$

Where $X_{Gaussian}$ denotes the Gaussian vector of the same covariance matrix as X. If the variable is Gaussian, the J(X) is zero, and generally, is non-negative. The Negentropy is nonparametric and expensive in computations, although it is robustness in statistic considerations. Thus, the negentropy is very difficult in the computation. Hence,

different approximations have been introduced for calculating the negentropy [18]. The negentropy can be approximated by higher-order cumulants, such as given in Equation (2.13) [58].

$$J(X) = \frac{1}{12}\,(E(X^3)^2) + \frac{1}{48}\,(\mathrm{kurt}(X)^2) \ldots (2.13)$$

Where X random variable is assumed to have zero mean and unit variance, kurt(.) is the kurtosis defined in Equation(2.9), and $E(.)$ represents the expected value of the random variable.

2.9 FastICA Algorithm

Based on a fixed-point iteration strategy, the FastICA algorithm extracts the independent components through maximizing the non-Gaussianity for the extracted signals by maximizing the negentropy [60]. It has convergence speed is at least quadratic, making it considerably faster than algorithms based on Gradient, that have linear convergence. Also, it is very easy to apply since the algorithm does not contain any values that need to be determined beforehand, such as the learning coefficient [70].

FastICA may be used to extract one independent component (IC) that is referred to extraction of one-unit. FastICA finds the weight vector w that extracts one independent component. A learning rule updates the values of w by searching for a direction that maximizes non-Gaussianity. This procedure named deflation approach, while FastICA

can estimate all independent components simultaneously in this state called symmetric approach.

FastICA is one of most popular linear ICA method, it depend on the fixed-point iteration method [71, 72]. Which will use in this dissertation for text clustering. The basic model of this algorithm is called the one-unit model. The basic formula of FastICA algorithm can be represented in algorithm (2.1) as follows:

Algorithm (2.1) : FastICA Algorithm

Input : Input data X_w % centering and whitening data

Output: Independent Component %recovered sources

Begin

1. Determine m % m is the number of independent components to be estimated.
2. Randomly,choose an initial value of weight vector w.
3. calculate $w^o = E(x_w g(w^T x_w)) - E(g'(w^T x_w))w$
4. Perform the normalization $w = w^o/\|w^o\|$
5. If convergence cannot be achieved , go back step 4.
6. $S=W\times X_w$

End Algorithm.

Where g represents the function used in the approximation to negentropy, g′ is derivative of g function. The purpose of orthogonal zing vectors is to prevent various vectors from convergence to the same maximum point. It is important to note that convergence means that the old and new values of w point in the same direction.

2.10 Metaheuristic Optimization Algorithms (MHOAs)

Metaheuristic algorithms are becoming an important part of modern optimization. A wide range of metaheuristic algorithms have emerged over the last two decades, and many Metaheuristic such as particle swarm optimization are becoming increasingly popular [73].

Most Metaheuristic algorithms are nature-inspired [74, 75], such as ant colony optimization [76], particle swarm optimization (PSO) [19], and cuckoo search algorithm [77]. Several new Metaheuristic algorithms have been produced after the emergence of swarm intelligence approaches such as the PSO that appear in the 1990s, and these techniques have been used in practically different fields of optimization, data mining, scheduling and planning, machine intelligence, and many more.

Exploration and exploitation are the two main components of the MHOAs. The first is the diversification process, that involves the searching process or exploring the entire search space to find new better solutions that are more diverse, and the second process is the intensification, which involves exploiting the information found in

current better solutions in the local search region, this information will be processed by iterations under certain conditions to produce the optimal solution. Selection, crossover, and mutation are evolutionary operators that affect the exploration and exploitation process to produce high-quality solutions [78].

Some MHOAs are found to be good in exploration and others in exploitation during the searching process. The exploration helps for global optimum while exploitation for local optimum. Therefore, the good optimization algorithm can maintain a proper trade-off between exploration and exploitation while maintaining its efficient search behavior to find "global" most optimal solution. This also helps the algorithm to avoid getting trapped in local optimum and premature convergence.

Also, some of the MHOAs use memory to keep track of the search process and find the optimal solution based on the previous solutions stored in the memory but others are found to be memory-less. The MHOAs are mostly categorized into swarm intelligence based, physics phenomenon based, evolutionary and others. Based on the search process, the MHOAs are categorized as: single solution-based or population-based algorithms depending on the search procedure. Single solution-based method begins with one candidate solution and improves it through iteration. Whereas, in population-based search, start with a set of candidate solutions that are improved through iterations, and finally the one with good fitness is selected as the optimal solution [79, 80].

The process of minimizing or maximizing an objective function or fitness function for certain constraints in order to produce a fitness value or objective value for making the system operate effectively is referred to as optimization [78]. Several nature-inspired Metaheuristic optimization algorithms (MHOA) have been developed and successfully used for the optimization of machine learning approaches such as artificial neural networks (ANN), support vector machines (SVM), and so on.

In this dissertation, proposed the ICA algorithms are based on Metaheuristic algorithms to propose a system for text clustering. PSO and GSO are population methods and use memory to find the optimal solution. PSO has several advantages are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques [81]. GSO algorithm enables a swarm of agents to split into subgroups, it is automatically divide swarm into subgroups which can then converge to multiple global optima simultaneously, this property of the algorithm allows it to be used to identify multiple peaks of a multi-modal function [82].

2.11 Swarm Intelligence (SI)

Swarm Intelligence, which is the collective behavior of self-organized particles, is commonly observed in nature. Swarm intelligence methods use a technique dependent on a search that employs a

distributed approach in which each agent acts independently [83]. These agents work with their neighbors to learn more about the environments. Those agents in this approach work in the following two stages: First stage deals with the agents performing an exploration behavior. While exploring, they seek data and check if it is above a fixed pre-defined threshold. These agents broadcast the data they have gathered to their neighbors by different communication channels (Lucifer in in case of glow-worm [84], pheromone in case of ants [85], etc.). This data is received by other agents in the swarm's neighborhood range. If the agent finds a value greater than the predefined threshold, it changes from exploration case to search case. If not, the agent will continue with the exploration behavior until it receives or senses a data value exceeds the pre-defined threshold.

In the search phase, to find the optimal data source, the agents start to collaborate with their neighbors. In order to continue the search, each agent uses its own data and data received from its neighbors to find a promising direction to move. If it is greater than its own sensing value, the agent switches its search direction towards the position of the agent which provides the maximum and hence more promising value in its neighborhood. If not, the agent continues the search in its current path. On collaborating with surrounding agents, after detecting data values above the threshold, the agent becomes a member of a virtual team that is exploring a particular promising area of the environment which in turn

leads to the autonomous emergence of different teams of cooperating agents, which is the main philosophy behind swarm intelligence [86].

2.11.1 Particle Swarm Optimization (PSO)

Kennedy and Eberhart [19] suggested the PSO method, which is a Metaheuristic method based on the concept of swarm intelligence capable of solving mathematical problems. It is necessary to note that, when compared to other optimization methods, the working with PSO produces some advantages compared with other optimization techniques, since there are fewer parameters to adjust during operation [87].

In order to explain how the PSO had inspired the formulation of an optimization algorithm to solve complex mathematical problems, a discussion on the behavior of a flock is presented. The (PSO) algorithm [19] is dependent on the behavior of birds flocking. The search agents are the birds' flock (also known as particles) that makes the algorithm's population. To prevent colliding, a flock of birds searching for food maintains a safe distance between them while flying. The flock's birds exchange information about the food source between them. The best value (pbest) is remembered by each bird. They adjust their velocity based on their positions during sudden movements and change of directions to maintain the flock movement. Moreover, each bird remembers the global best position of one bird who discovered the greatest value (gbest). Finally, the flock of birds finds the food source at the global best position on movement after the maximum number of

iterations specified. The best optimum solution is referred to as the bird with gbest.

In computational science, PSO is a technique of computing that improves a problem by iteratively refining a candidate solution to a given measure of quality. It solves a problem by generating a population of possible solutions, known as particles, and moving these particles in the search region using mathematical formulae that describe the particle's position and velocity. Considering a swarm with P particles, there is a position vector $X_i^t = (x_{i1}\ x_{i2}\ x_{i3}\ \dots\ x_{in})^T$ and a velocity vector $V_{ti} = (v_{i1}\ v_{i2}\ v_{i3}\ \dots v_{in})^T$ at (t) iteration for each one of the (i) particle that composes it. These vectors are updated through the dimension j according to the following equations:

$$V_{ij}(t+1) = WV_{ij}(t) + c_1 r_1 \left(pbest_{ij}(t) - X_{ij}(t)\right) + c_2 r_2 \left(gbest_i(t) - X_{ij}(t)\right) \dots . (2.14)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \qquad \dots \dots \quad (2.15)$$

Where i = 1,2,…,P and j = 1,2,…,n.

Equation (2.14) represents movement of a particle in an iteration, thus it has several terms that will be presented. The (*w*) is the inertia weight parameter, which is a positive constant value in the classical PSO version. This parameter is important for balancing the global search, also known as exploration (when higher values are set), and local search, known as exploitation (when lower values are set). The second term of

Equation (2.14), is the individual cognition term which is computed by difference between the best position of particle $pbest_{ij}$, and current position X^t_{ij} of this particle. The parameter (c_1) existing in this term is a positive constant and it is an individual-cognition parameter, and it weighs the importance of particle's own previous experiences. The other parameter that makes up the second term's product is (r_1), which is a random value parameter with a range of [0,1]. This parameter is essential because it prevents premature convergences and increases the likelihood of global optima [19].

Finally, social learning is the third term. It allows all particles in the swarm to exchange information about the best location they've found, regardless of whose particle discovered it. The difference (gbest - X^t_{ij}) attracts particles to the best position until it is discovered at iteration (t). Similarly, (c_2) is a parameter of social learning that evaluates the significance of the swarm's global learning. And (r_2) serves the same purpose as (r_1). In the meanwhile, the positions of the particles are updated using Equation (2.15).

2.11.2 Glowworm Swarm Optimization(GSO)

GSO is a swarm intelligence algorithm that was introduced by Krish-nanand and Ghose in 2005 [88]. It is essentially developed for numerical optimization problems that require calculating multiple optima of multimodal functions, as against to other swarm intelligence techniques which aim to determine the global optimum. Each worm in the swarm is attracted with and migrates toward one neighbor when the

glow of this neighbor is brighter than this worm itself. When this worm is surrounded by many such neighbors, it applies a probabilistic approach to select one of them. Each glowworm contains luciferin, a luminous amount that allows it to communicate with its neighbors and exchange information.

GSO is extremely good to capturing the global optimum of the objective function in finite-dimensional vector space and can successfully prevent losing the optimal solution due to intelligent changes in the decision radius. In the GSO algorithm, initially ,a swarm of glowworms are distributed in the solution area, randomly. Each glowworm carries a specific amount of luciferin and indicates a solution to the objective function in the search space. The luciferin amount is associated with the fitness of the current position of the agent. Specifically, Using a probabilistic method, each agent can only be attracted by a neighbor whose luciferin intensity is higher than its own within the local-decision domain and then moves towards it. The density (amounts) of a glowworm's neighbors affects its decision radius and determines the size of its local-decision domain: when the neighbor-density is low, the local-decision domain will enlarge in order to find more neighbors; otherwise, it will reduce to allow the swarm to split into smaller groups. The previous steps are repeated until the algorithm satisfies the termination condition. At this moment, the majority of individuals gather around brighter glowworms. The GSO includes five major stages which will describe briefly:

The first stage is luciferin-update: The luciferin update is dependent on both fitness and previous luciferin values [89], and it follows the rule given in the following Equation.

$$l_i(t+1) = (1-\rho)\, l_i(t) + \gamma \text{Fitness}(x_i(t+1)) \ldots\ldots\ldots (2.16)$$

$l_i(t)$ indicates the luciferin value of glowworm i at time t, ρ is the luciferin decay constant, γ is the luciferin improvement constant; $x_i(t+1) \in R$ is the glowworm location i at time $t+1$, and Fitness($xi(t+1)$) describes the value of the fitness at glowworm i's position at time $t+1$.

The second stage is neighborhood-select: neighbors $N_i(t)$ of glowworm i at t time, consist of the brighter ones and can be described as in Equation (2.17):

$$N_i(t) = \{j : d_{ij}(t) < r^i_d(t);\ l_i(t) < l_j(t)\} \ldots\ldots\ldots (2.17)$$

Where $d_{ij}(t)$ denotes the Euclidean distance between glowworms i and j at time t, and $r^i_d(t)$ denotes the radius of decision (local decision) of glowworms i at time t.

The third stage is moving probability: A glowworm moves towards other glowworms with greater luciferin levels using a probability rule. The probability $P_{ij}(t)$ indicates the moving of glowworm (i) towards its neighbor j can be described as follows:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad \ldots\ldots\ldots (2.18)$$

The fourth stage is movement: If there are two glowworm (i) and (j) . Also that (i) selects (j) with consider $j \in N_i(t)$ with $P_{ij}(t)$. The equation of movement i toward j will be as:

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{||x_j(t) - x_i(t)||}\right) \ldots\ldots\ldots (2.19)$$

Where, $\| \cdot \|$ indicates the operator of Euclidean norm, and (s) is indicating to the step size.

The fifth stage is decision radius-update: the glowworm decision radius (i) at each updated, is represented as follows:

$$r_d^i(t+1) = \min\{r_s, max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \ldots\ldots (2.20)$$

Where β indicates a constant, r_s indicates the glowworm sensory radius (i), and n_t represents the parameter to control the number of neighbors. The decision radius(local decision) and the sensory radius of glowworm (i), can be displayed in Figure 2.10.

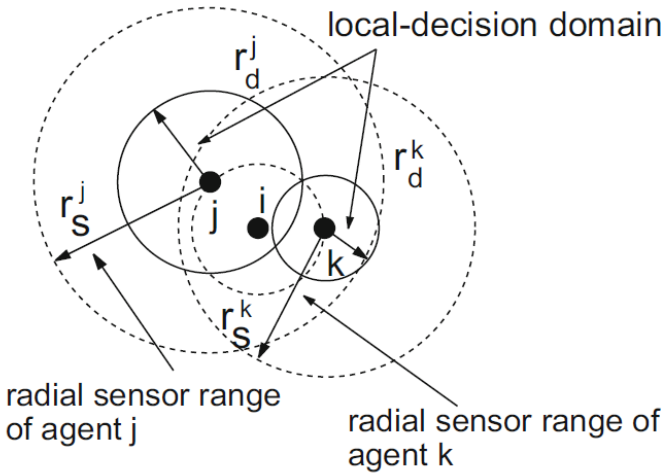


Figure 2.10: Sensory and Decision Radial of Three Glowworms i,j,and k [20].

2.12 Argmax Function

The mathematical function argmax is widely used in machine learning applications. It is an operation that finds the argument that gives the maximum value from a target function. It's also the most popular method for identifying the cluster with the highest expected probability in machine learning [90]. Argmax is a function that takes a vector Z of n real numbers as input and returns the index of the maximum value of a vector as Equation (2.21).

$$\text{Result} = \text{argmax}(Z) \ldots\ldots (2.21)$$

Where Z represents a vector, the result is the index of high value in a vector. After carrying out Equation (2.5) of FastICA and proposed

algorithms, the sources will be obtained which represent the vectors with elements where each one represents a sample (text). Argmax function will calculate the probability of belonging the text to a certain IC which represents the cluster.

2.13 Evaluation of Text Clustering

Cluster validation is a term used to describe the process of assessing the quality of clustering algorithm outputs. Clustering validation statistics can be divided into three categories:

1- Internal cluster validation, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information.

2- External cluster validation, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels.

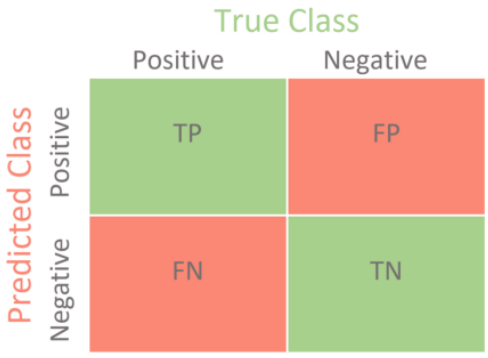
3- Relative cluster validation, which evaluates the clustering structure by varying different parameter values for the same algorithm [91].

The second evaluation paradigm, which is called external validation. It is the style of evaluating clustering results by adding external information. The labeled examples are prepared as the test collection, their labels are hidden during clustering, and the similarity

between items is based on their label consistencies during the evaluation [91].

In general, the evaluation measures in clustering and classification problems are defined from a matrix with number of examples correctly and incorrectly classified for each category, this matrix named confusion matrix for classification and matching matrix for clustering. The matching matrix for the clustering problem for two classes is shown in Table 2.2.

Table 2.2: The Matching Matrix



The following are descriptions of the TP, FP, FN, and TN concepts:

1- True Positive (TP): examples are those that are correctly predicted to belong to the positive class.

2- False Positive (FP): examples that are predicted to be positive but are actually negative.

3- False Negative (FN): examples that are predicted to be negative but are actually positive.

4- True Negative (TN): instances correctly predicted as belong to the negative class.

The main criteria for evaluating the effectiveness of the system are precision (P), recall (R), F-measure, and overall accuracy. The fraction of recovered instances that are relevant to the search is known as precision. The precision refers to the accuracy of model. It's also known as the positive prediction value [92, 93]. The typical precision equation is provided in Equation 2.22.

$$P = \frac{TP}{TP+FP} \quad \ldots \quad (2.22)$$

The recall is the percentage of relevant examples recovered out of all relevant examples [92, 93]. The typical recall equation is as follows:

$$R = \frac{TP}{TP+FN} \quad \ldots. \ (2.23)$$

The F-measure is a precision-to-recall trade-off. It is the harmonic mean value of both precision and recall [92, 94]. The conventional F-measure equation is as follows:

$$F_{Measure} = \frac{2P \times R}{P+R} \quad \ldots\ldots\ldots \ (2.24)$$

CHAPTER THREE

TEXT CLUSTERING SYSTEM

CHPTER THREE

TEXT CLUSTERING SYSTEM

3.1 Introduction

A major amount of the variety of information currently is underwritten via news, blog, and social networking sites available on the internet. With the increasing volume of data that available, there is a need for electronic systems to process and deal with this data.

The proposed system and its stages were described in this chapter. A brief description of the proposed system is given in section 3.2. A detailed definition of the suggested methods and function are given from section 3.3 through section 3.8.

3.2 Proposed System Overview

The proposed system is aims to design and develop a clustering system for text data using ICA machine learning algorithm, and implementation Metaheuristic algorithms to improve the performance of the ICA algorithm by used negative entropy function as the objective function. In this context, the ICA unsupervised machine learning method has been used for building and developing text clustering of two types of datasets, which requires several steps include resource obtaining (datasets), preprocessing of text, text representation, and ICA machine learning algorithms and clustering.

The structure of the proposed system will be discussed in details through display the diagram of system. Figure (3.1) displays a block diagram of the suggested system.

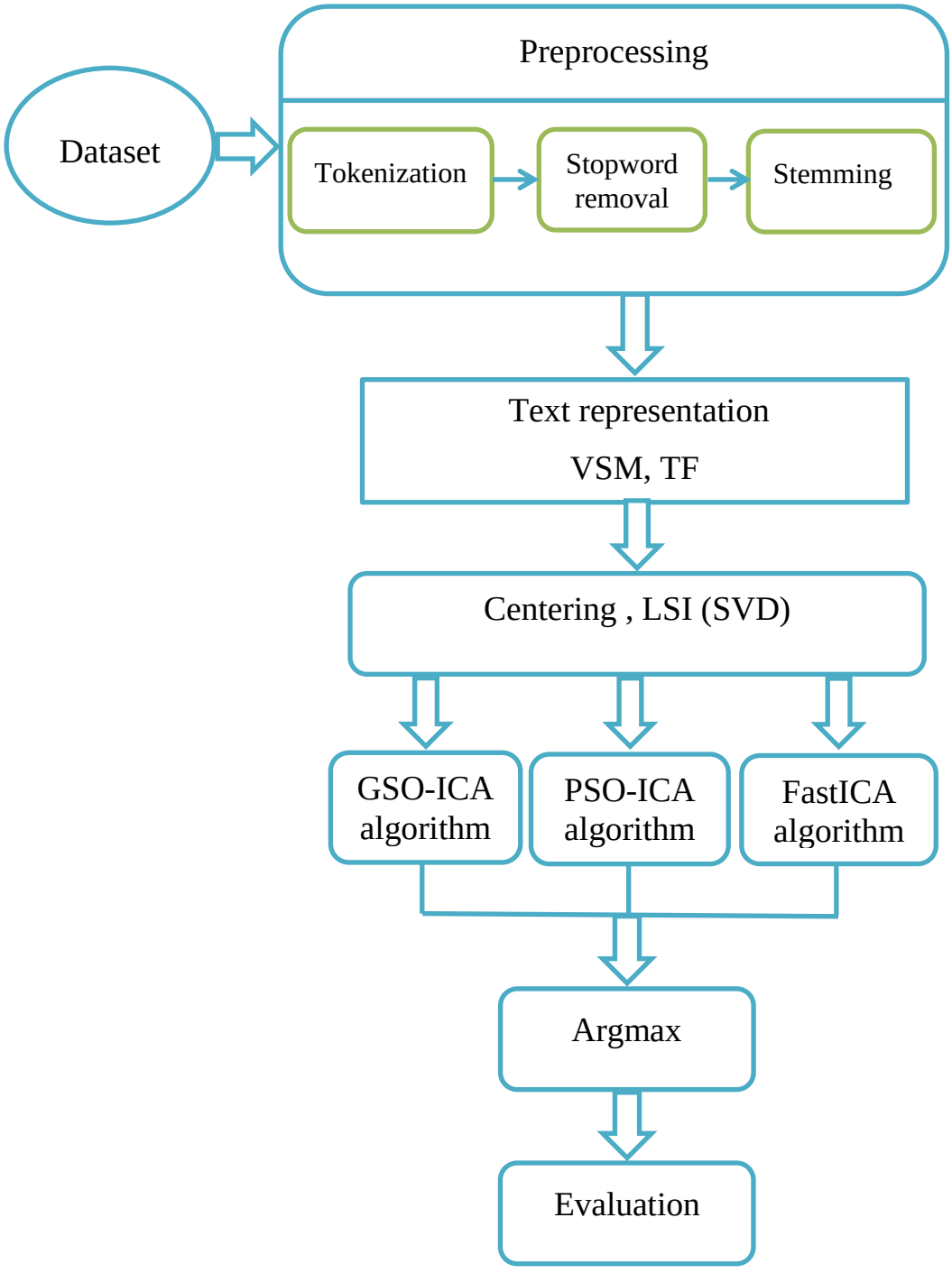


Figure 3.1: Block Diagram of the Proposed Text Clustering System.

3.3 Initialize Data

As know, the fundamental concept of ICA algorithms is to isolate variables (sources) of known observations (which are assumed to be a linear combination of the unknown sources). The interpretation is the documents can be inferred as mixtures of latent concepts grouping terms (terms refer to latent concepts). Where the concepts demonstrate a succinct and informative representation of the themes in the text. This is meaning that the concepts are defined as units of knowledge.

This dissertation claims that using concepts as the basis of clustering can significantly improve effectiveness. Therefore we will employ the two datasets are medical dataset and BBC news dataset. After converted text datasets into numerical data, and presents this data as an input matrix, such as a term-by-document matrix to the proposed system of text clustering. Applied the FastICA algorithm, and two proposed ICA based on Metaheuristic algorithms to the perform the clustering system. The overall steps for the proposed text clustering system illustrates in general algorithm (3.1).

Algorithm (3.1) overall steps for the proposed text clustering system.

Input : Set of documents.

Output: Clusters for the documents.

1- Preprocessing

- Tokenization %call algorithm 3.2.

- Stopword Removal %call algorithm 3.3.
- Stemming %call algorithm 3.4.

2- Text representation

- Dictionary building algorithm %call algorithm 3.5.
- Converting texts to a matrix %call algorithm 3.6

3- Preprocessing ICA algorithm

- Centering %call algorithm 3.7.
- Whitening %employ SVD technique as in section 2.6.3 , call algorithm (3.8).

4- Proposed Model and Optimization Methods

- Clustering system by using PSO-ICA and GSO-ICA. % call algorithm 3.9 and algorithm 3.10, these algorithms call algorithm 3.11 to compute the fitness function.

5- Calculate the probability that the text belongs to a certain cluster.

- Argmax algorithm %call algorithm 3.13.

3.4 Preprocessing

Text preprocessing phase includes tokenization, stop words elimination, and stemming.

3.4.1 Tokenization Algorithm

In most languages, the text is composed of words divided by whitespace. In this dissertation, the work will be with English language of two different datasets. The steps of tokenization processing display in algorithm (3.2):

Algorithm (3.2) :Tokenization Algorithm

Input : Set of documents (D)

Output : Tokens (T_i)

Begin

1. Input documents are collected as (D_i), where i=1,2,3,….,n.
2. For each input D_i:

T_i = tokenize(D) % For each input document D_i, cutting word by using the white space as the delimiter process for all documents this process it uses, i=1, 2, 3…n.

EndFor

End Algorithm

3.4.2 Stopword Removal Algorithm

To achieve the accuracy of text analytics applications, it is important to filter out the redundant terms of low or no significance. This process is referred to as stopwords removal. It is a traditional and simple method based on removing stop words by compared the words of

document with the words were stored in the list stopwords file so if there is any match, the word will be removed from the document. The steps of stopword removal are displayed in the algorithm (3.3).

Algorithm (3.3) : Stopwords Removal Algorithm

Input : Set of tokens for each text, and stopwords list .
Output : Words vector(V) for each text without stopwords.

Begin

For each text D_i:
V[i]=[ ]
For each token(T_j):
If T_j in stopwords list:
Remove T_j
Else
Insert Tj into V[i]
EndIf
EndFor
EndFor

End Algorithm.

3.4.3 Stemming Algorithm

Using the Porter stemming method, a particular suffix on the word is removed by applying the set of rules. These rules are

implemented when a set of criteria meet, thereby having precisely matching stems. This procedure minimizes the number of words and saving memory space and time. Algorithm (3.4) show the steps of stemming processing.

Algorithm (3.4) : Stemming algorithm

Input : Word vector (V_i) for each text without stop words .

Output : Stemmed word vectors (Dc) % each vector represents document.

Begin

1. Read word vector (V).
2. Stem the words % using the porter stemming technique.
3. Output the stemmed word (write to a file).
4. Repeat step 2 and step 3 until reach end of the document.

End Algorithm

3.5 Text Rrepresentation Algorithms

After carrying out the preprocessing steps described in section 3.4. The important next step is to create a dictionary from the documents (Dc) which obtained from the preprocessing stage. The dictionary will be building from all the unique words in the Dc. The steps of building dictionary presents in algorithm (3.5).

Algorithm (3.5) : Dictionary building Algorithm

Input : set of documents (Dc) % The documents resulted from algorithm (3.4)

Output : Dictionary of words (Fw) % dictionary

Begin

1. For each text D_c
2. For each word F in D_c
3. If F not in Fw
4. Add F into Fw
5. End If
6. End For
7. End For

End Algorithm.

After building a dictionary, applied one of the most important stages of text clustering is text representation. Its aim is to numerically describe unstructured text documents so that they can be handled mathematically. Algorithm (3.6) display the steps of converting dataset documents to the numeric matrix.

Algorithm (3.6) : Converting documents to matrix

Input : Dc % a set of N documents .

Fw % a set of M words (dictionary)

```
Output :  X          % matrix with  M  rows and  N  columns .
 Begin
   i=1          % Initialize value of counter dictionary
   j=1          % Initialize value of counter documents

   1.  For each document Dj in Dc          % main loop iteration for
                                                each document.
       Read document Dj

   2.    For each word in Fw    % main loop iteration for each word
             s= number of occurrence Fwi (word) in Dj
             If  (s > 0) then
                   X[i][j]=s
             Else
                   X[i][j]=0
             End If
       End For
     End For

End Algorithm.
```

The result of the algorithm (3.6) is the term-document matrix (X) which is formatted as shown in Equation (2.1). The next stage beyond calculating the frequency of a words in a document is to modify the count by the perceived importance of that words. Equation(2.2) has been applied to calculate weighting or importance for words.

3.6 Preprocessing of ICA Algorithms

Before carrying out FastICA, and proposed algorithms, must conduct two common steps are centering, and whitening to prepare data. These two phases of the preprocessing are indeed preprocessing operations that optimize the subsequent ICA core processing. In the following subsections, the algorithms of these steps are present.

3.6.1 Centering Algorithm

Centering is usually conducted at the beginning step before whitening. It would make computation easier. As mentioned in section (2.6.2), "Procedure of Learning ICA". In this step, the mean(X_{mean}) is subtracted from the observation data X. In this chapter, the centering process steps are displayed in the algorithm(3.7).

Algorithm (3.7) : Centering Algorithm

Input: Input data(X) % vectors of matrix

Output: Zero mean vectors (X_C) % centered vectors

Begin

1. Calculate the mean of observed data(vector)

   $X_{mean} = \frac{1}{n}\sum_{i=1}^{n} X_i$ % mean of vector *X* with length *n*

   where n represent the number of columns.

2. Subtract X_{mean} from the observed vector X

   $X_C = X - X_{mean}$

End Algorithm.

3.6.2 Whitening Algorithm

The whitening process is the second important pre-processing step in the ICA algorithm. This process is mentioned in detail in section (2.6.3). The proposed system used the SVD technique as the whitening process. The algorithm (3.8) presents the whitening steps of data.

Algorithm (3.8) : Whitening Algorithm

Input: Centering data (X_C) % centering vectors

Output: Whitening data (X_W) % Whitening Vectors of matrix

Begin

1. Calculate the SVD of centering data(X_C) % calling SVD technique.

   X_C =SVD(X_C) %The result is three matrices are U,$\sum$,V^T

2. Calculate multiply eigenvalue matrix $\sum$ with eigenvector V^T matrix.

   $X_W = \sum V^T$ % X_W represents (concept-document) matrix

End Algorithm.

3.7 Proposed Model and Optimization Methods

At this point, obtained data represent the concept-document matrix where each document is the combination of concepts. These data are separated by using ICA algorithms. In this dissertation, the proposed

two Metaheuristic optimization based ICA methods, and the objective function that will be used are discussed in the next section.

3.7.1 Proposed Optimization Methods

The text clustering method is at the essence of the proposed system. Two Metaheuristic optimization algorithms (PSO and GSO) are proposed as the optimization algorithms for the linear mixtures of the traditional ICA algorithm. Negentropy was employed as an objective function in proposed PSO-ICA and GSO-ICA algorithms. The results which will be seen in the fourth chapter of this dissertation are compared to the standard FastICA algorithm (2.1) which mention in chapter 2.

3.7.1.1 PSO- ICA algorithm

In this method, the Particle Swarm Optimization (PSO) algorithm suggested as an improving method employs negative entropy (which is based on kurtosis) as an objective function to improve the performance of the ICA method. This method is shown as in the algorithm (3.9).

Initialization of the necessary parameters of the PSO algorithm to be appropriate for the ICA algorithm.

Algorithm (3.9) : PSO-ICA Algorithm

Input: Input data X_w % centering and whitened data
Output: Independent Component % recovered sources

% Variables Definition

K: number of sources % integer variable

population: population size % integer variable

v_i : velocity of the particles % real variable

x_i: position of the particles % real variable

w : inertia weight value.

r_1,r_2 : two uniform random numbers % uniform numbers

c_1,c_2 : two acceleration coefficients

v_{max}: maximum velocity value of current particle

$-v_{max}$: minimum velocity value of current particle

Pglobalmax:maximum global fitness value of particle

%real variable

Plocalmax: local fitness value of particle % real vector

fitnessmax: list of maximum local fitness values % real vector

fglobalmax: list of maximum global fitness values % real vector

fitness: initial fitness values of current positions of particle

%real vector.

fitnessnew : list of next fitness values % real vector

iteration: max number of iterations=10. % integer number

Begin

1. Randomly generating a set of particles % each particale represent demixing matrix.

```
W=np.random(population, K, K)
```

2. Initializing the velocities of the particles

v = np.random(population K, K)

3. Compute initial fitness values of the current positions of particles to the objective function.

For i =1 to population

S= $W \times X_w$

% The proposed objective function

fitness(i) = sum(negentropy function) % calculate current fitness value

End For

4. Set initialize values of the algorithm parameters% local values

plocalmax= fitness

5. Capture the initial maximum value of the fitness value.

pglobalmax = maximum (fitness)

6. Main loop iterations of the algorithm for each particle.

iter=0 % iteration index

For n in range(0,population):

For i in range(0,K):

For j in range(0,K):

$v[n,i,j] = vmax \times v[n,i,j] + c_1 \times r_1 \times (plocalmax[n,i,j] - W[n,i,j]) + c_2 \times r_2 \times (pglobalmax[i,j] - W[n,i,j])$

% Evaluate v[n,i, j] with the (vmax)and (-vmax)parameters

W[n,i,j]=W[n,i,j]+v[n,i,j] % update the position

EndFor

EndFor

EndFor

7. Compute a new fitness values of the positions of particles

For m in range(0,population):

$y = W \times X_w$

% call the proposed objective function

fitnessnew(m)=sum(negentropy fun.) % calculate a new fitness value.

EndFor(m)

8. Set maximum values of *fitnessnew* and *pglobalmax* as maximum fitness values and its positions respectively.

9. Find a new maximum value of the fitness values.

pglobalmax = max (fitnessnew)

10. Increment the iteration and stopped

Iter=+ 1 % Increment the iteration counter.

Until iter = iteration % Terminate the loop of PSO algorithm

11. S = y % sources retrieved

End Algorithm.

The algorithm (3.9) represents the ICA method based on the PSO algorithm. Following the initialization of the algorithm's parameters, the first step is to randomly generate the separating matrix, followed by the initialization of the particle velocities in the second step. Then, computed the initial values of the fitness function using the proposed objective function in the third step. In the fourth step, initialize values for the algorithm's parameters. The main loop of the PSO algorithm is used in steps 5-9 before the predefined iteration number is reached, the algorithm uses the suggested objective function to

determine the fitness values for each iteration within this loop and, it continues proceeds to the next steps in order to maximize the fitness values and, finding the best W matrix based on global position, which is used to compute the sources according Equation (2.5).

3.7.1.2 GSO-ICA algorithm

One of the most effective optimization methods is Glowworm Swarm Optimization (GSO), with quick search processes and fast convergence without getting stuck in local minima states. GSO algorithm is used to improve the performance of the ICA method and suggested a way of the ICA based on the GSO, as shown in the algorithm (3.10).

Algorithm (3.10) : GSO-ICA Algorithm

Input: Input data X_w %centering and whitened data

Output: Independent Component %recovered sources

% Glowworm swarm parameters

K: number of sources

itermax: Iterations number

m : Dimensions number

n : Number of glowworms

r_s: Radial sensor range

r_d: Decision radius % (r_s*ones(n,1))

γ: Luciferin enhancement factor

ρ: Luciferin value decay factor

s : Step size % distance of worm movement

β: Constant parameter

n_t: Parameter to control the neighbor number

bound:The workspace range parameter

Li: Indicates the luciferin value of glowworm . % (5*ones(n,1))

Begin

1. Initialize the algorithm parameters with their initial values.
2. A group of (n) glowworms are distributed randomly.

   A=rand(n, m)
3. Main loop of the algorithm.

   set initial value of the counter (t =1).

   Itermax= 100.

   While (t <= itermax):

   3.1. Luciferin Update Phase:

   -Set Initialize Separated Matrix from the distributed worms

   $W_{i,j}= A_{i,n}$

   -Initialize separated signals

   $S = W\times X_w$

   3.2. For each Glowworm i

   Fitness(x_{t+1}) = Negentropy Function

   $L_{t+1}=(1- \rho) L_t+ \gamma$Fitness($x_{t+1}$) % call algorithm 3.11.

   3.2.1. Neighborhood Select Phase: % for each Glowworm i,

   Neighbors consist of the brighter ones

$N_i(t) = \{j: d_{ij}(t) < r^i_d(t); L_i(t) < L_j(t)\}$ % as in eq. (2.17)

3.2.2. Moving Probability Phase: probability of glowworm (i) going towards its neighbor (j):

$$\boldsymbol{p_{ij}(t) = \frac{L_j(t) - L_i(t)}{\sum_{k \in N_i(t)} L_k(t) - L_i(t)}}$$ % as in eq. (2.18)

3.2.3. Movement Phase: If glowworm(i) selects glowworm(j), $j \in N_i(t)$ with $P_{ij}(t)$; movement of glowworm(i).

$$\boldsymbol{A_i(t+1) = A_i(t) + s\left(\frac{A_j(t) - A_i(t)}{||A_j(t) - A_i(t)||}\right)}$$ % as in eq. (2.19)

3.2.4. Decision Radius Update: calculate the glowworm's radius:

$r^i_d(t+1) = \max\{0, \min\{r_s, r^i_d(t) + \beta(nt - |N_i(t)|)\}\}$

% as in eq.(2.20)

4. Glowworms have new values now.

t=t+1 % increase the counter of iterations

5. Get new separated matrix W from the A matrix that satisfy the maximum value of luciferin (L).

$W_{i,j} = A_{i,n}$

6- compute the sources(S) by using ICA equation (2.5).

$S_{best} = W \times X_w$

End Algorithm.

The algorithm (3.10) represents the ICA method based on the GSO algorithm. This algorithm employed the negentropy function as

the objective function. Through the steps of the algorithm, a demixing matrix was obtained (W). Afterward, by using the W matrix, the sources matrix (S) is obtained by applied the Equation (2.5) in final step.

3.7.2 Fitness Function

The ICA method depends essentially on the optimization algorithm and the objective function. This section includes description of the function that proposed as the objective function which can be used in the proposed algorithms as described in the section (2.7). In this section, will discuss the *Negentropy* function to be used as an objective function in the proposed ICA algorithms. The algorithm of this function has been described in next subsection.

3.7.2.1 Negentropy Algorithm

This function represents the approximation of the negentropy function based on the fourth-order cumulants (*kurtosis*) consider best objective function which used in the proposed system as shown in chapter Two. This function described in Algorithm (3.11).

Algorithm (3.11) : Negentropy Algorithm

Input: *S* % unmixed data (real vector)
Output: *neg* % negentropy value of S

Begin

1. Calculate the mean of the vector S

$$\mathbf{m_{ean}} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{S_i}$$ % mean of vector S with length n.

2. Calculate the Kurtosis of the vector *(S)* by using the algorithm (3.13) .

$K(S) = kurt(S)$.

3. Calculate the *negentropy* according to the eq. (2.13)

$$\mathbf{neg} = \frac{\mathbf{1}}{\mathbf{12}} (\mathbf{m_{ean}}(\mathbf{S})^{\mathbf{3}})^{\mathbf{2}} + \frac{\mathbf{1}}{\mathbf{48}} (\mathbf{kurt}(\mathbf{S})^{\mathbf{2}})$$ % Kurt : kurtosis

End Algorithm.

Algorithm (3.12) represents the *kurtosis* of a vector is as follow:

Algorithm (3.12) : Kurtosis algorithm

Input: S % unmixed data (real vector)

Output: kurt % the value represent the kurtosis of S

Begin

1. Calculate the mean of the vector S

$$\mathbf{m_{ean}} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{S_i}$$ % mean of vector (S) with length n

2. Calculate the Kurtosis of *S* according to the eq. (2.9)

$kurt = m_{ean}(S^4) - 3 \times (m_{ean}(S^2)^2$

End Algorithm.

3.8 Argmax Algorithm

At this stage, the proposed ICA algorithms revealed separate components (ICs), each of which represents a cluster. The argmax function calculates the probability of a document belonging to a certain cluster based on a maximum value. The algorithm (3.13) represents the steps of Argmax function.

Algorithm (3.13) : Agrmax Algorithm
Input: Matrix of S % matrix with dimension 5×n
Output: value of index % index represent the position of high probability value.

Begin
For j : 1 to n: %n represent number of columns(texts) in vector
For i: 1 to 5: % vector represent the values of 5 sources
index = index of maximum S[i,j] % return the index of a maximum value which indicates the relevance of the particular text to a particular IC(cluster).
EndFor
EndFor

End Algorithm.

The Argmax function, according to the algorithm, answers not how large the maximum is, but where it happens. Generally, a higher value indicates more relevance of the concept to a particular document. $S_{concept\text{-}document}$: it assigns a measure of how relevant a concept is given a particular document based on the value of concept.

CHAPTER FOUR

EXPERIMENTAL RESULTS

CHAPTER FOUR

EXPERIMENTAL RESULTS

4.1 Introduction

In this chapter, the methodology used to assess the proposed system is introduced. When machine learning is used, it needs to be tested by using an independent dataset to ensure it functions as intended. In this assessment, two datasets in were used. Several experiments were conducted to evaluate the performance of the text clustering system. Due to unsupervised machine algorithms are used in this system, labeled objects could be used to assess clustering algorithms by comparing to help determine the groups and get more meaningful results.

Two algorithms proposed are ICA based on the PSO algorithm (PSO-ICA), and ICA based on the GSO algorithm (GSO-ICA). These algorithms used the Negentropy as an objective function. The results of these algorithms compared with the standard FastICA algorithm as an optimum example of the traditional ICA algorithms.

The proposed algorithms and evaluation measurements were programmed under Python code and evaluated using recall, precision, F-measure and Overall Accuracy (Macro-average F) metrics.

4.2 Experimental Datasets

Obviously, gathering data is the first step of text mining (i.e., the relevant documents). The related documents may already be provided or may be part of the problem description in certain text mining scenarios. If the documents were identified previously, then they can be obtained

directly, and the main issue is to cleanse (preprocess) the samples and ensure that they are of high quality.

Such as with non-textual data, Human interference could negatively impact the credibility of the process of document gathering, As a result, severe caution is needed. Sometimes, documents can be acquired from document warehouses or databases.

Two standard datasets are used to demonstrate and assess the proposed algorithms. The medical abstract dataset and the BBC news dataset were acquired from ftp://ftp.cs.cornell.edu/pub/smart/ and http://mlg.ucd.ie/datasets/bbc.html respectively.

The first dataset is a medical abstract (MED) dataset, which comprises 1,033 texts that have 30 labels. This dissertation utilized subsets including 124 abstracts annotated in five sets. The groups from the first group to fifth included 37, 16, 22, 23, and 26 documents respectively. Figure 4.1 displays the distributed text documents inside medical dataset.

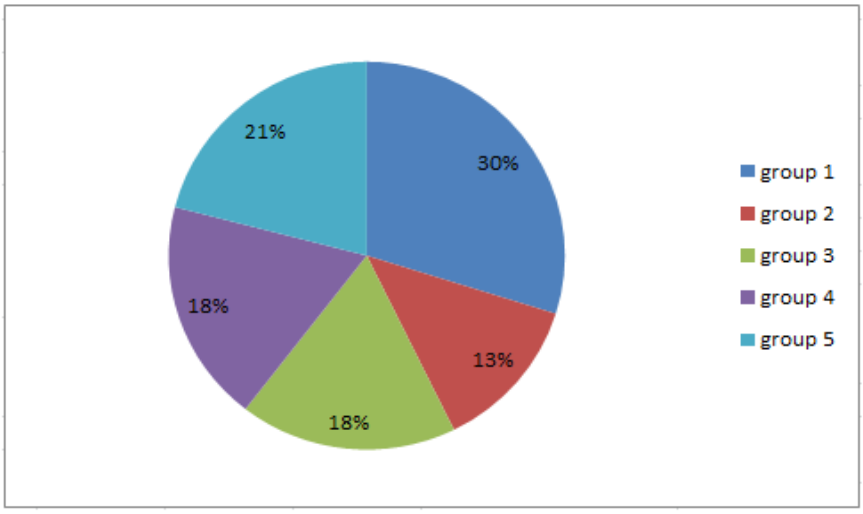


Figure 4.1: The Percentage of Distribution of Documents Inside Medical Dataset

The second dataset acquired from the BBC news website contains 510, 386, 417, 511, and 401 documents in five topical areas that are business, entertainment, politics, sport, and tech respectively. Figure 4.2 displays the distributed documents inside the BBC news dataset.

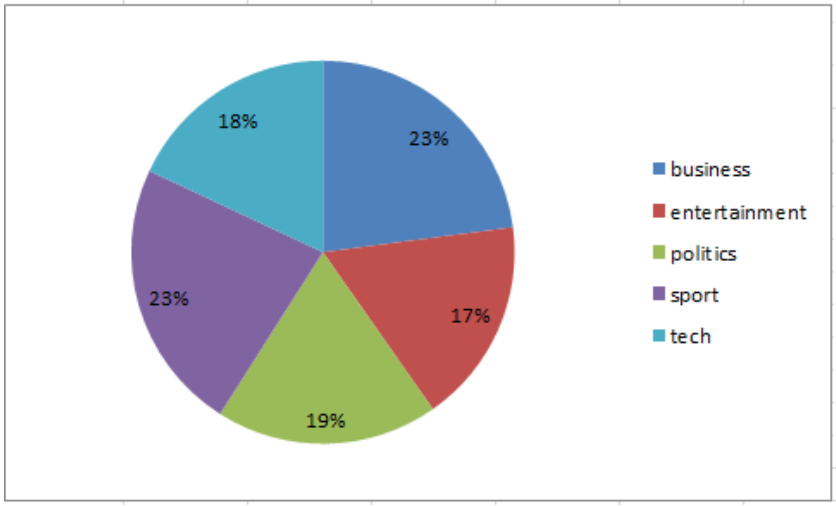


Figure 4.2: The Percentage of Distribution of Documents Inside BBC News Dataset

Document clustering encountered many challenges. Such as datasets comprise high-dimensional with respect to words, therefore documents are sparse and have varying lengths, at the same time can contain correlated terms. Thus, constructing a model to represent a document based on the concept can be used to distinguish between documents, as a solution to the clustering task. The clustering algorithms to calculate the similarity between documents and assessment is highly dependent on the chosen document model. In this dissertation, an unsupervised ICA algorithm provides the statistical model where each document is represented by one value. Therefore, external cluster validation will be applied, which will include comparing the outcomes

of cluster analysis to an externally known result. It measures the degree to which cluster labels matching to cluster labels that have been provided externally.

4.3 Preparing Data

After conduct preprocessing of datasets, the term-document matrix X was built, when creating the term-document matrix, these terms that appeared in several documents and were not included in a defined list of stop words as stop words. To eliminate the influence of document length, the length of all texts (columns) was standardized to one length based on the length of the BOW list (dictionary). After that, carried out the centring process of the data that was used as the inputs of the LSI procedure, where the singular value decomposition SVD was used as a whitening process. As shown in Equation (2.7), the matrix (X) had been decomposed by SVD. In order to construct the $D = \sum V^T$ matrix given in Equation (2.8), the highest eigenvalues of the diagonal matrix $\sum$ were taken into consideration.

The largest principal components of matrix D were used as inputs of the standard FastICA algorithm, and the two proposed PSO-ICA and GSO-ICA algorithms for computing the five ICs, which were used as clusters for clustering the documents. To quantify the ability of the ICs to cluster documents we convert the separated signals (ICs) into “clusters probabilities” using the argmax function. The estimated IC cluster label for a given document or sample matches the document with the component number (index) with the highest probability.

4.4 Interpretation of the Components

The estimated components ICs are the result of applying ICA algorithm to the matrix D; it represents the concept-document matrix. Documents could be inferred to be mixtures of latent concepts. Therefore, each row (signal) represents the mixtures of concepts and each column represents the text (sample) in the signal.

ICA algorithms will extract the independent signal (IC) which will contain number of values equal to number of documents, where each value represent one concept, this concept expresses a brief description of the text.

4.5 Experimental Results of Medical Dataset

In this dissertation, conducted the clustering experiments on the medical documents. Three experiments were performed to show the effectiveness of the suggested system.

4.5.1 First Experiment

The MED dataset consists of 124 documents. This means that the mixed signals are of length 124 samples (texts). Five signals were taken from the D matrix based on large eigenvalues as inputs to the FastICA algorithm.

The FastICA algorithm will start with an initial value of the separating matrix and employ the logcosh function to update the separating matrix in an iterative manner to find the best value for the

separating matrix to achieve the best separation of the sources. Five separate signals were obtained, each of which represents an independent component IC with a length of 124 samples. Then the probability of the sample (text) belonging to one of the components is calculated based on the largest value that depicts the concept using the argmax function.

The MED dataset was used to evaluate the performance of the FastICA method. The experiment was done utilizing the five IC components. When converting IC (recovered sources) to clusters using the argmax function we also matched the unsupervised ICA clusters to the manual labels. The matching matrix compared the outcomes of the FastICA clusters with the annotated texts by experts manually. Table 4.1 present the matching matrix of the FastICA method with each text cluster.

Table 4.1: Matching Matrix of The FastICA Method in each Text Cluster in First Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	**31**	**0**	**1**	**0**	**0**
IC2	**0**	**16**	**2**	**1**	**0**
IC3	**5**	**0**	**18**	**5**	**0**
IC4	**1**	**0**	**1**	**17**	**2**
IC5	**0**	**0**	**0**	**0**	**24**

Table 4.1, displays the matching matrix of the MED dataset clustering based on the five IC components of outcomes of the FastICA clusters. Where each column in the matrix represents the actual cluster

and each row represents the predicted cluster. For example, the 31 represents the predicted value is positive and it is true (TP) for cluster 1 from all amount 37 texts , and the rest values in all columns represent the predicted value for cluster 1 which is positive but false (FP) which equal 1 text. Whereas, represents the rest all rows in column 1 the predicted values which are negative but it is positive (FN) which is equal 6 texts.

The standard assessments were applied to evaluate performance. Popular examples of these metrics include recall, precision, F-measure, and macro-average. Based on a matching matrix that includes true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), these metrics are calculated. Table 4.2 will show the precision, recall, and F-measure of the FastICA experiment according to the Equations (2.22, 2.23, and 2.24).

Table 4.2: Overall Accuracy of The FastICA Method in The First Experiment

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	**0.969**	**0.838**	**0.903**
Cluster 2	**0.842**	**1.0**	**0.921**
Cluster 3	**0.643**	**0.818**	**0.731**
Custer 4	**0.810**	**0.739**	**0.774**
Cluster 5	**1.0**	**0.923**	**0.962**
Overall Accuracy			**0.855**

This analysis demonstrates that good outcomes were achieved by FastICA, thereby proving that the FastICA method is suitable for implementing the text clustering application. As in Table 4.2, the overall F-measure of 85.5% was achieved for text clustering.

4.5.2 Second Experiment

After set initial values of the parameters of the proposed PSO-ICA algorithm that are mentioned as shown in chapter three. It applied the proposed algorithm and the negentropy function on the same data in the MED dataset. Using the matching matrix, the results were compared to the documents that had been manually annotated. The matching matrix of the PSO-ICA algorithm is shown in Table 4.3 for each text cluster.

Table 4.3: Matching Matrix of The PSO-ICA Method in each Text Cluster in The Second Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	**37**	**0**	**0**	**0**	**0**
IC2	**0**	**16**	**0**	**1**	**0**
IC3	**0**	**0**	**16**	**4**	**0**
IC4	**0**	**0**	**6**	**18**	**2**
IC5	**0**	**0**	**0**	**0**	**24**

Table 4.3 exhibits the matching matrix of the MED dataset clustering based on the five ICs components by the comparison of the outcomes of the PSO–ICA method with the annotation MED dataset.

To compute the accuracy of the PSO-ICA algorithm, using the metrics precision, recall and, F-measure. Table 4.4 shows the percentage of metrics of the results of the method.

Table 4.4: Overall Accuracy of The PSO-ICA Method in The Second Experiment

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	**1.0**	**1.0**	**1.0**
Cluster 2	**0.941**	**1.0**	**0.971**
Cluster 3	**0.80**	**0.727**	**0.764**
Custer 4	**0.692**	**0.783**	**0.737**
Cluster 5	**1.0**	**0.923**	**0.962**
Overall Accuracy			**0.895**

The result of the second experiment explains that the proposed PSO-ICA method is superior to the standard FastICA algorithm, where producing an overall accuracy of 89.5% for text clustering as shown in Table 4.4.

4.5.3 Third Experiment

After set initial values of the parameters of the proposed GSO-ICA algorithm that are identified as shown in chapter three. It applied to the same data in the MED dataset. Also, using the matching matrix, the results were compared to the documents that had been manually

annotated. The matching matrix of the GSO-ICA algorithm is displayed in Table 4.5 for each text cluster.

Table 4.5: Matching Matrix of The GSO-ICA Method in each Text Cluster in The Third Experiment.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
IC1	**37**	**0**	**2**	**0**	**0**
IC2	**0**	**16**	**0**	**1**	**0**
IC3	**0**	**0**	**16**	**3**	**0**
IC4	**0**	**0**	**4**	**19**	**2**
IC5	**0**	**0**	**0**	**0**	**24**

Table 4.5 shows the matching matrix of the MED dataset clustering based on the five ICs components by the comparison the outcomes of the GSO–ICA method with the annotation MED dataset.

To assess the performance of the GSO-ICA method, using the standard metrics to calculate the accuracy of the suggested algorithm. Table 4.6 shows the percentage of metrics of the results of the method.

Table 4.6: Overall Accuracy of The GSO-ICA Method in The Third Experiment

	Precision (%)	Recall (%)	F-measure (%)
Cluster 1	**0.949**	**1.0**	**0.974**
Cluster 2	**0.941**	**1.0**	**0.971**
Cluster 3	**0.842**	**0.727**	**0.785**

Custer 4	**0.760**	**0.826**	**0.793**
Cluster 5	**1.0**	**0.923**	**0.962**
Overall Accuracy			**0.903**

The third experiment demonstrates that the GSO-ICA algorithm provides a good result with an accuracy of 90.3% for text clustering, as shown in Table 4.6. It is superior to FastICA and providing results higher than the PSO-ICA algorithm.

PSO-ICA and GSO-ICA are two algorithms were proposed to use negentropy as objective function to improve the performance of traditional ICA algorithm. The results of these algorithms compared with the standard FastICA algorithm as good example of the traditional ICA algorithms. Figure 4.3 shows values of accuracy achieved for text clustering of MED dataset.

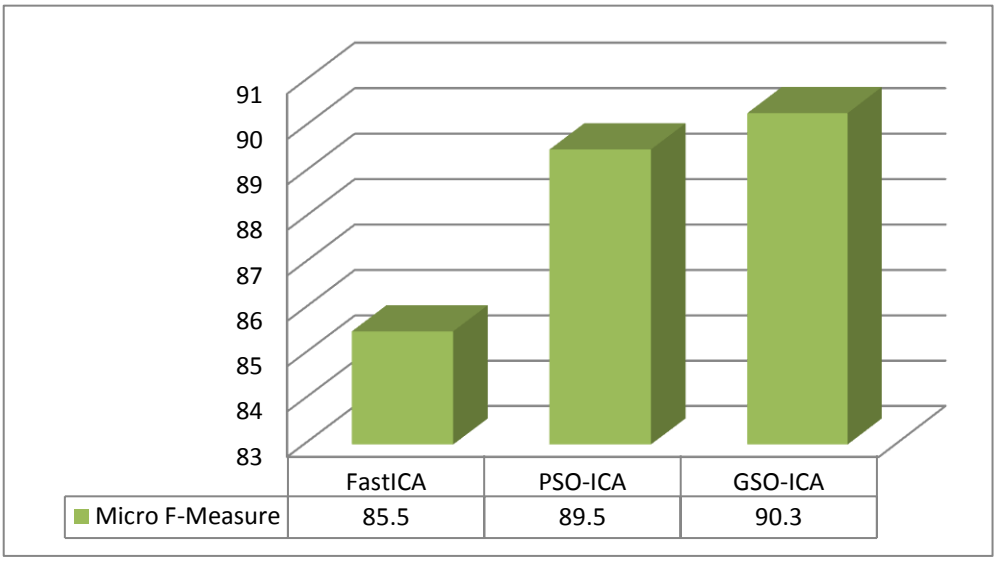


Figure 4.3: The Accuracy Achieved for Text Clustering of MED Dataset.

Figure 4.3 shows the results obtained by the FastICA, PSO-ICA, and GSO-ICA algorithms. According to the experiments of text clustering of MED dataset, the highest result yielded by GSO-ICA with 90.3% accuracy, and the lowest results yielded by FastICA with 85.5% accuracy. We also note that the PSO-ICA algorithm achieved accuracy superior to the FastICA algorithm by 89.5%.

These results depict the outperformance of the proposed algorithms compared with the standard FastICA algorithm. Because of the GSO-ICA and PSO-ICA algorithms can possible to escape from local minimum and provide optimal solution. In addition, the experimental results showed that the GSO-ICA superior on PSO-ICA due to ability to split agents into subgroups and find multiple global solution simultaneously.

4.6 Experimental Results of BBC Dataset

This experiment aimed to verify the clustering system correctness percentage when we used other datasets different in size and type of distribution of documents among the clusters as input texts to the clustering system.

Two main experiments are executed by FastICA, PSO-ICA, and, GSO-ICA algorithms which applied to the BBC news dataset. In these experiments, the first one takes 250 documents, 50 documents for each group. The second one is applied to the whole BBC news dataset which contains 2225 documents.

4.6.1 First Main Experiment

Includes three implicit experiments which are applied on a subset documents of the BBC news dataset comprise 250 files, divided 50 documents for each group.

4.6.1.1 Fourth Experiment

The subset documents of BBC news dataset consist of 250 documents. This means that the mixed signals are of length 250 samples (documents). Five signals were taken from the D matrix based on a large eigenvalues as inputs to the FastICA algorithm.

After applied the FastICA algorithm, five separate signals will be getting, each of which represents an independent component IC with a length of 250 samples. Then the probability of belonging sample to one of the components is calculated based on the largest value that represents the concept using the Argmax function.

The FastICA algorithm's performance was evaluated using 250 manually labelled BBC news documents. The five ICs were used in the experiment. The matching matrix-matched FastICA results to hand-annotated texts. Table 4.7 shows the FastICA matching matrix for each text cluster.

Table 4.7: Matching Matrix of The FastICA Method in each Text Cluster in Fourth Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	**45**	**4**	**1**	**0**	**5**
IC2	**0**	**40**	**1**	**0**	**2**
IC3	**3**	**6**	**47**	**0**	**6**
IC4	**0**	**0**	**0**	**49**	**0**
IC5	**2**	**0**	**1**	**1**	**37**

Also, the typical metrics were used to assess the performance of the algorithm. Table 4.8 shows the percentage of metrics of the results of the FastICA method to the subset BBC news dataset.

Table 4.8: Overall Accuracy of The FastICA Method in The fourth Experiment.

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	0.818	0.9	0.859
Cluster 2	0.93	0.8	0.865
Cluster 3	0.758	0.94	0.849
Custer 4	1.0	0.98	0.99
Cluster 5	0.902	0.74	0.821
Overall Accuracy			0.872

These results show that FastICA produced a good outcomes, were produced an overall F-measure 87.2%. These values show that the algorithm gives higher results when the number of documents is equal inside the corpus.

4.6.1.2 Fifth Experiment

PSO-ICA algorithm was applied in this experiment, where the parameters of the algorithm are installed. It was applied on the same subset BBC news dataset, and the matching matrix was used to compare the findings to the manually annotated texts. Table 4.9 shows the PSO-ICA method's matching matrix for each text group.

Table 4.9: Matching Matrix of The PSO-ICA Method in each Text Cluster in The Fifth Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	**45**	**5**	**3**	**0**	**3**
IC2	**0**	**38**	**0**	**1**	**2**
IC3	**3**	**6**	**45**	**0**	**6**
IC4	**0**	**0**	**0**	**48**	**0**
IC5	**2**	**1**	**2**	**1**	**39**

Table 4.10 displays the proportion of metrics of the findings that demonstrate the method's effectiveness in terms of performance.

Table 4.10: Overall Accuracy of The PSO-ICA Method in The Fifth Experiment

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	0.898	0.88	0.889
Cluster 2	0.935	0.86	0.897
Cluster 3	0.833	1.0	0.917
Custer 4	0.961	0.98	0.97
Cluster 5	0.932	0.82	0.876
Overall Accuracy (Macro-F-Measure)			0.908

As indicated in Table 4.10, the second sub experiment shows that the PSO-ICA algorithm outperforms the standard FastICA method, with an average F-measure of 90.8% percentage for text clustering when compared to the standard FastICA method which provided 87.2%.

4.6.1.3 Sixth Experiment

GSO-ICA algorithm was applied in this experiment on same subset BBC news dataset. Through the use of the matching matrix, the outcomes were compared to the values of the humanly annotated texts. Table 4.11 exhibits the matching matrix of the GSO-ICA method in each text cluster.

Table 4.11: Matching Matrix of the GSO-ICA Method in each Text Cluster in The Sixth Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	**45**	**0**	**0**	**1**	**2**
IC2	**1**	**47**	**4**	**0**	**4**
IC3	**3**	**1**	**46**	**0**	**2**
IC4	**0**	**1**	**0**	**49**	**0**
IC5	**1**	**1**	**0**	**0**	**42**

Table 4.12 displays the percentage of measurements of the data that indicate the method's efficiency, according to the standard metrics (precision, recall and, F-measure).

Table 4.12: Overall Accuracy of the GSO-ICA Method in the Sixth Experiment

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	0.938	0.90	0.919
Cluster 2	0.839	0.94	0.89
Cluster 3	0.885	0.92	0.902
Custer 4	0.98	0.98	0.98
Cluster 5	0.955	0.84	0.897
Overall Accuracy (Macro-F-Measure)			0.916

As seen in Table 4.12, the third sub experiment shows that the GSO-ICA algorithm produces a successful result for text clustering, with

an accuracy of 91.6% percentage. It is outperforms FastICA and outperforms the PSO-ICA algorithm in terms of performance.

According to the results of the experiments, PSO-ICA and GSO-ICA improved the accuracy of the standard ICA algorithm. The accuracy of these algorithms as compared to the standard FastICA algorithm for text clustering of a subset of the BBC news dataset as seen in Figure 4.4.

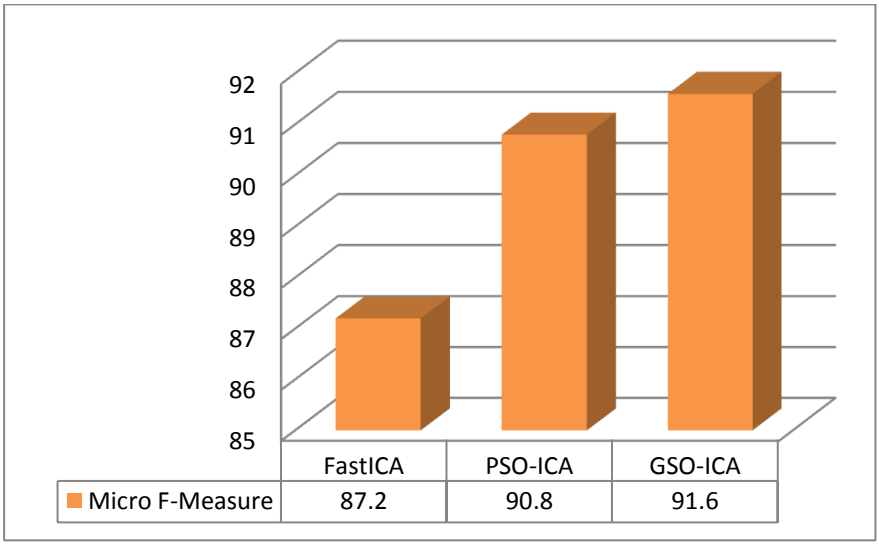


Figure 4.4: The Accuracy Achieved for Text Clustering of Subset BBC News Dataset.

The performance of the PSO-ICA and GSO-ICA algorithms is seen in Figure 4.4. According to the findings of text clustering tests on the subset BBC dataset, GSO-ICA produced the best results with 91.6% percentage accuracy, while FastICA produced the lowest results with 87.2% percent accuracy. Also, we have seen that the PSO-ICA

algorithm outperformed the FastICA algorithm in terms of accuracy with 90.8%.

4.6.2 Second Main Experiment

Include three implicit experiments which were applied on a whole of BBC news dataset which comprises 2225 documents.

4.6.2.1 Seventh Experiment

In this experiment, the standard FastICA algorithm was applied. The FastICA algorithm's output was evaluated using 2225 manually labelled BBC news documents from the corpus.

The whole BBC news dataset consists of 2225 documents. This means that the mixed signals are of length 2225 samples (documents). The inputs of algorithm will be five signals taken from D matrix. The results will be five separate signals with length 2225 equal to number of documents in corpus. Then the probability of the belonging document to one of the components is calculated based on the largest value that depicts the concept using the Argmax function. It compared the outcomes of FastICA algorithm with the manually annotated documents from the entire BBC news dataset, which served as the basis for the matching matrix. In each text cluster, the FastICA method produces a results, which is shown in Table 4.13.

Table 4.13: Matching Matrix of the FastICA Method in each Text Cluster in The Seventh Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	**458**	**18**	**42**	**2**	**14**
IC2	**3**	**289**	**0**	**9**	**4**
IC3	**30**	**25**	**358**	**74**	**12**
IC4	**2**	**25**	**13**	**422**	**10**
IC5	**17**	**29**	**4**	**4**	**361**

The percentage of results measurements that suggest the method's reliability is seen in Table 4.14.

Table 4.14: Overall Accuracy of the FastICA Method in The Seventh Experiment.

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	0.858	0.898	0.878
Cluster 2	0.948	0.859	0.788
Cluster 3	0.717	0.859	0.788
Custer 4	0.894	0.826	0.86
Cluster 5	0.87	0.90	0.885
Overall Accuracy (Macro-F-Measure)			0.849

This experiment shows that FastICA produced good results when applied to a large dataset, indicating that the method is appropriate for accomplishing the text clustering application. Text clustering yielded an

average overall accuracy of 84.9% percentage, as seen in Table 4.14. This finding shows that when the algorithm is used for various datasets of varying sizes, its output remains constant or converges.

4.6.2.2 Eighth Experiment

PSO-ICA algorithm was used in this experiment. After initialize the parameters of the algorithm .It was applied on whole BBC news dataset. Also, the manually annotated documents were used for comparison. Table 4.15 shows the PSO-ICA method's matching matrix for each document cluster.

Table 4.15:Matching Matrix of the PSO-ICA Method in each Text Cluster in the Eighth Experiment.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
IC1	**٤٧٢**	**11**	**6**	**2**	**9**
IC2	**2**	**323**	**3**	**5**	**12**
IC3	**27**	**31**	**399**	**17**	**10**
IC4	**1**	**11**	**4**	**487**	**46**
IC5	**8**	**10**	**5**	**0**	**324**

Typical metrics were used to assess the performance of the algorithm. Table 4.16 explains the percentage of metrics found by the PSO-ICA algorithm on the whole BBC news dataset.

Table 4.16: Overall Accuracy of the PSO-ICA Method in The Eighth Experiment

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	0.944	0.925	0.935
Cluster 2	0.936	0.837	0.887
Cluster 3	0.824	0.957	0.891
Custer 4	0.887	0.953	0.92
Cluster 5	0.934	0.808	0.871
Overall Accuracy (Macro-F-Measure)			**90.1**

The results show when applied PSO-ICA to a large dataset, it yielded reasonable performance, confirming that the proposed algorithm is proper for performing the text clustering employment. As seen in Table 4.16, text clustering produced an overall F-measure of 90.1% percentage. This result demonstrates that the algorithm's performance stays stable or converges when applied to different datasets of differing sizes.

4.6.2.3 Ninth Experiment

This experiment utilized the GSO-ICA algorithm. After initializing the algorithm's parameters. All of the BBC news dataset was subjected to the GSO–ICA algorithm and compare the results with the human-annotated texts using the matching matrix. GSO-ICA method

produces a matching matrix for each document cluster as shown in Table 4.17.

Table 4.17:Matching Matrix of the GSO-ICA Method in each Text Cluster in the Ninth Experiment.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
IC1	478	39	17	1	10
IC2	1	283	0	5	13
IC3	13	28	386	7	4
IC4	6	19	4	494	5
IC5	12	17	10	4	369

Also, the standard measurements were used to evaluate the system's efficiency. The percentage of metrics of the GSO-ICA method's findings on the whole BBC news dataset is seen in Table 4.18.

Table 4.18: Overall Accuracy of the GSO-ICA Method in The Ninth Experiment.

	Precision (%)	**Recall (%)**	**F-measure (%)**
Cluster 1	0.877	0.937	0.907
Cluster 2	0.937	0.733	0.835
Cluster 3	0.881	0.926	0.903
Custer 4	0.936	0.967	0.951
Cluster 5	0.896	0.92	0.908
Overall Accuracy (Macro-F-Measure)			0.903

The third sub experiment as seen in Table 4.18, demonstrates that the GSO-ICA algorithm achieves a great outcome for text clustering, with an accuracy of 90.3% percentage. In terms of efficiency, it outperforms both FastICA and the PSO-ICA method.

The results reveal that PSO-ICA and GSO-ICA enhanced the performance of the traditional ICA method. Figure 4.5 shows the performance of these algorithms relative to the standard FastICA algorithm for text clustering through the entire BBC news dataset.

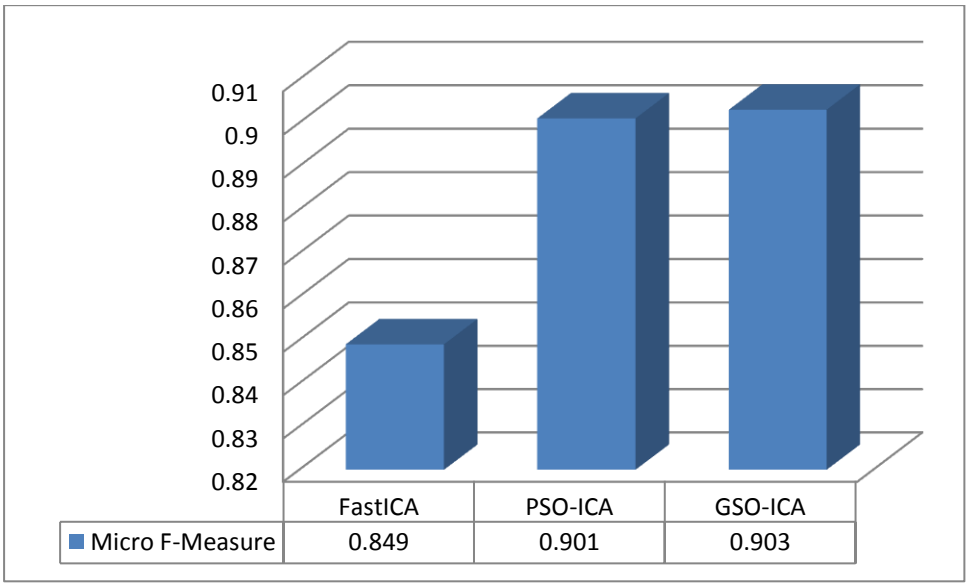


Figure 4.5: The Accuracy Achieved for Text Clustering of Entire BBC News Dataset.

Figure 4.5 depicts the success of the PSO-ICA, and GSO-ICA algorithms. GSO-ICA reported the best results with 90.3% percentage accuracy, while Fast-ICA produced the lowest results with 84.9%

percentage accuracy, according to the results of text clustering experiments on the entire BBC dataset. In addition, the PSO-ICA algorithm outperformed the FastICA algorithm by 90.1% percentage in terms of accuracy.

Table 4.19 shows summary of experimental datasets, including the name of the dataset, number of documents in each dataset, steps of preprocessing, number of clusters, and accuracy of algorithms.

Table 4.19:Details of All Experiments.

Dataset	Text documents	No. of clusters	Preprocessing			FastICA		PSO-ICA		GSO-ICA	
			Tokenize	stopword removal	stemming	accuracy	elapsed time	accuracy	elapsed time	accuracy	elapsed time
MED Dataset	١٢٤	٥	√	√	×	85.5%	0:00:06.927412	89.5%	0:00:07.222813	90.3%	0:00:08.328817
BBC news	250	5	√	√	√	87.2%	0:00:22.713640	90.8%	0:00:24.819643	91.6%	0:00:25.880445
BBC news	2225	5	√	√	√	84.9%	0:02:51.923489	90.1%	0:03:21.817554	90.3%	0:03:33.767175

In this chapter, analyzed the behavior of the proposed clustering methods. The started was by describing the corpus and then explain the evaluation methods so that we can explain the results of the experiments. In general, several experiments are carried on two different types of corpora with regard the size, type, and distribution of documents among the clusters.

The experimental results show that the proposed unsupervised clustering algorithms have a good accuracy and a stable performance approximately. Also they provides a high-grade results especially when take equal samples for each cluster from the BBC news dataset. In addition, the GSO-ICA algorithm produced the best results with highest elapsed time and FastICA algorithm produced the lowest results with lowest elapsed time for all experiments.

4.7 An Illustrative Example

In this section, a complete example of the process to find the independent components and compute the accuracy of the proposed system will be displayed. This example is start by conducting the preprocessing of data, then applying the algorithms, computing the probability of belonging the text to the cluster, and finally computing the accuracy of the system. The example will be conducted by using the subset BBC news dataset which include 250 files.

1- Starting by read the files from dataset. The Figure 4.6 represent the text document before preprocessing.

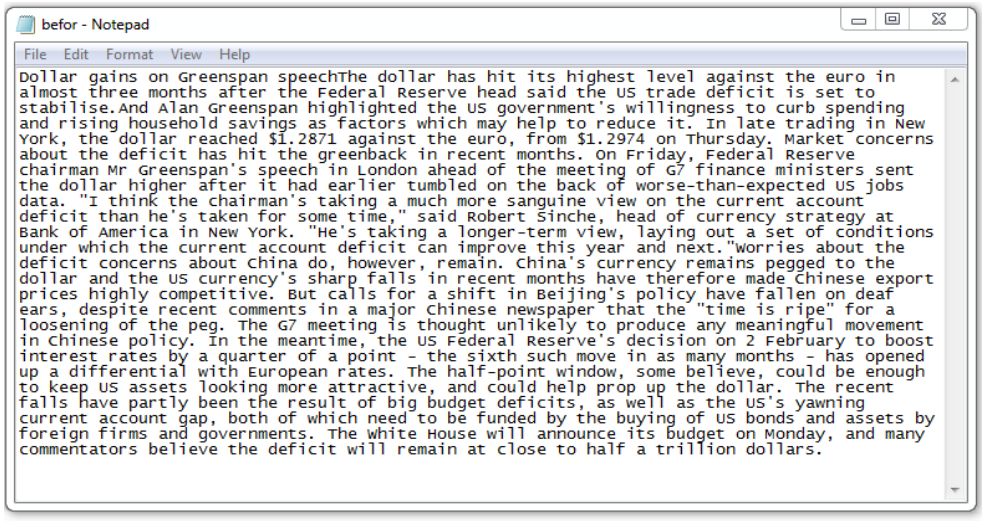


Figure 4.6: Text Document Before Preprocessing.

After implementing preprocessing will obtain the file, as shown in Figure 4.7. This file is free from stopwords and punctuation, and the words contained in the file represent the root of words.

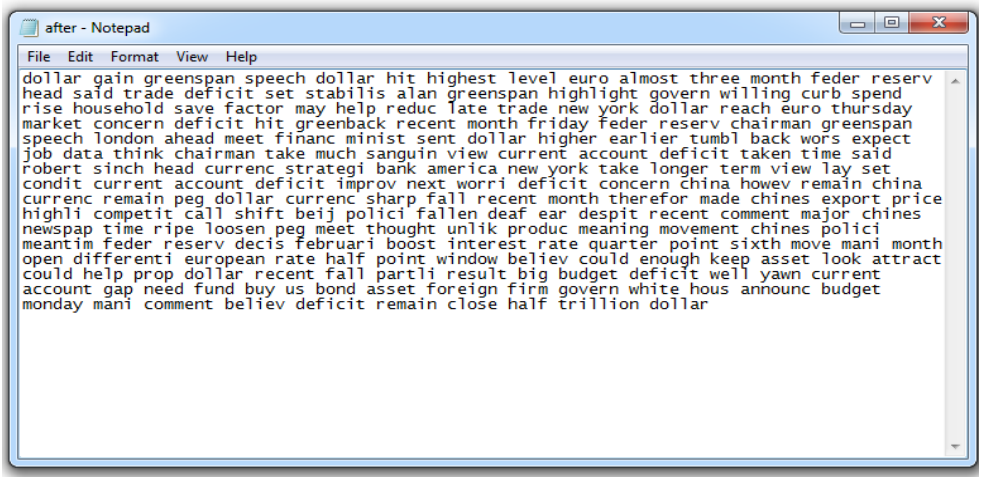
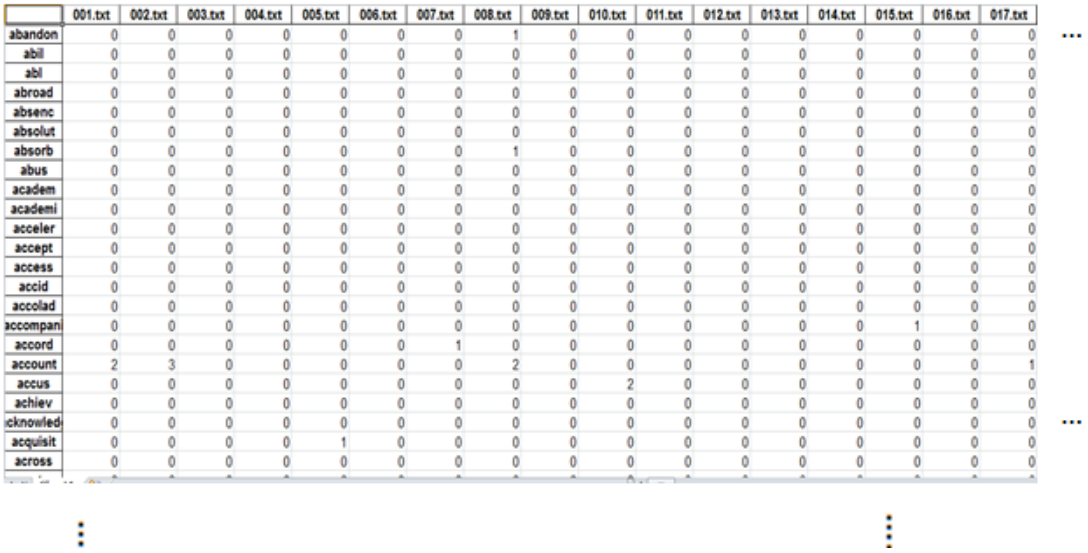


Figure 4.7: Text Document After Preprocessing.

2- Text representation: in this stage will create the dictionary, then converting texts to a matrix and compute the term frequency (TF) as display Figure 4.8.



	001.txt	002.txt	003.txt	004.txt	005.txt	006.txt	007.txt	008.txt	009.txt	010.txt	011.txt	012.txt	013.txt	014.txt	015.txt	016.txt	017.txt	…
abandon	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
abil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
abl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
abroad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
absenc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
absolut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
absorb	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
abus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
academ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
academi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
acceler	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
accept	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
access	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
accid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
accolad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
accompani	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
accord	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
account	2	3	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	
accus	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	
achiev	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
icknowled	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	…
acquisit	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
across	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 4.8: Term-Document Matrix.

After that compute the important of each word in text according the Equation (2.2), as Figure 4.9.

	001.txt	002.txt	003.txt	004.txt	005.txt	006.txt	007.txt	008.txt	009.txt	010.txt	011.txt	012.txt	013.txt	014.txt	015.txt	016.txt	017.txt
abandon	0	0	0	0	0	0	0	0.0062	0	0	0	0	0	0	0	0	0
abil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
abl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
abroad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
absenc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
absolut	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
absorb	0	0	0	0	0	0	0	0.0062	0	0	0	0	0	0	0	0	0
abus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
academ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
academi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
acceler	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
accept	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
access	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
accid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
accolad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
accompani	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0033	0	0
accord	0	0	0	0	0	0	0.0071	0	0	0	0	0	0	0	0	0	0
account	0.0096	0.0155	0	0	0	0	0	0.0125	0	0	0	0	0	0	0	0	0.0096
accus	0	0	0	0	0	0	0	0	0	0.0194	0	0	0	0	0	0	0
achiev	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
icknowled	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
acquisit	0	0	0	0	0.009	0	0	0	0	0	0	0	0	0	0	0	0
across	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.9: Term-Document Matrix After Computing the Importance of Words.

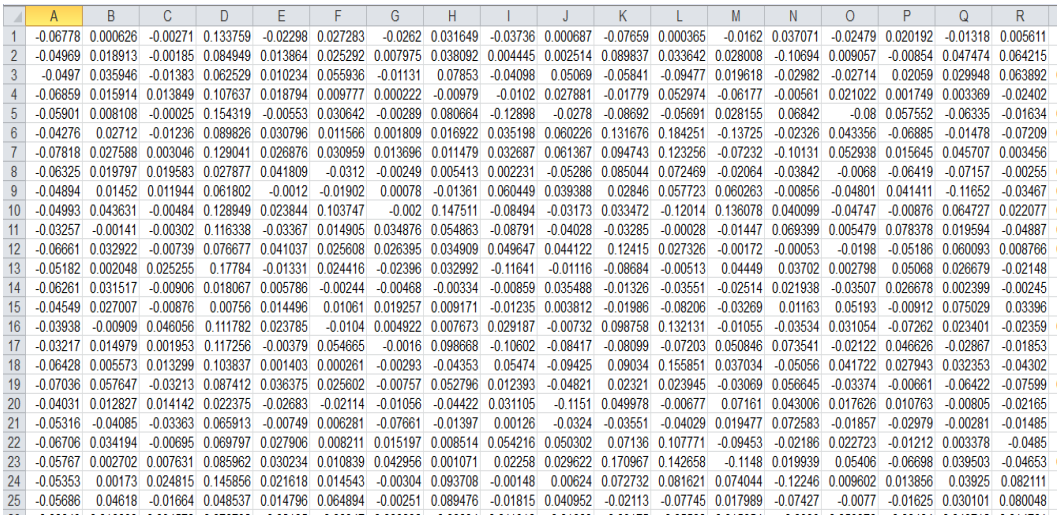
3- The data in the term-document matrix will be input into the ICA algorithm. Before applying the algorithm, the two preprocessing steps will be implemented which are centering and whitening. Figures 4.10, 4.11, 4.12 are the results of the SVD technique.

	A	B	C	D	E	F	H	I	J	K	L	M	N	O	P	Q	R
1	0.009592	0.003547	-0.00529	0.009316	0.008003	0.008176	0.008689	0.005935	0.009482	-0.00117	0.009218	0.004078	-0.00646	0.008794	0.008228	0.009132	-0.0113	-0.00563
2	0.002625	0.001775	0.002702	0.00252	0.000246	0.004533	0.002909	0.005223	0.002876	-0.03075	0.002842	0.006638	0.017005	-0.00049	-0.01231	0.003773	0.002842	0.007592
3	0.000506	0.00084	0.004606	0.000182	0.006972	-0.00236	0.000949	0.000773	-0.00071	-0.02475	0.000385	0.000524	-0.01145	0.008133	-0.00109	-0.00196	-0.01143	-0.00588
4	0.000101	-0.00149	0.003086	-0.00133	-0.00245	-0.00345	0.003892	-0.01089	-0.00178	0.003547	-6.1E-05	-0.01127	-0.00699	-0.00938	-0.00224	-0.00412	0.012579	0.026873
5	0.001384	-0.00651	-0.01406	0.002907	0.003224	0.005884	0.000561	0.006249	0.005934	-0.00039	0.002163	0.009365	-0.01657	-0.00346	-0.00169	0.003376	-0.01279	-0.01099
6	-0.00111	-0.00966	-0.01321	-0.00154	-0.00466	0.000575	-0.00191	0.00464	-8E-05	0.001281	-0.00041	0.000324	-0.02396	0.020546	0.005503	-0.0011	-0.00796	0.001133
7	-0.00116	-0.00133	-0.0051	-0.00043	-0.00357	0.000171	-0.00182	0.000242	0.001575	-0.00815	-0.00075	-0.00015	-0.00963	-0.00354	0.013756	0.000624	-0.00173	0.001535
8	-0.00177	0.013174	-0.01276	-0.00201	-0.01095	-0.00068	0.000591	-0.00193	-0.00318	-0.01431	-0.00107	-0.00085	0.001949	0.00247	0.002944	-0.0009	-0.00497	0.001144
9	0.001296	0.000534	0.020787	-0.00157	-0.00903	0.002683	0.000904	-0.00397	-0.00226	-0.00417	-0.00201	-0.0035	0.027887	0.002014	-0.00156	-0.00377	0.026238	-0.00812
10	-0.00119	0.007514	-0.00939	-0.00015	-0.00144	0.005427	-0.00012	0.000323	0.002507	0.009046	0.00125	-0.00589	-0.06384	0.000279	-0.0007	-0.00091	0.001745	-0.00269
11	0.002541	-0.00654	-0.01504	-0.00109	0.009201	0.001956	0.000943	0.006101	0.003027	-0.02204	-0.00351	-0.00325	0.015433	-0.00086	0.003167	-0.00331	0.01857	0.004025
12	-0.00129	0.003894	-0.00467	-0.0012	-0.01153	-0.00465	0.003489	-0.00689	0.004218	0.005807	-0.00317	-0.00052	0.004431	0.007834	-0.00415	0.001787	0.034717	0.017763
13	-0.00113	0.006871	-0.02109	-0.00069	-0.00556	-0.00391	-0.00727	0.015383	0.001607	-0.00713	-0.00112	-0.00516	-0.00855	-0.00974	0.001771	-0.00185	-0.02213	0.018047
14	-0.00037	-0.00654	-0.00672	0.001306	-0.00502	0.003985	-0.00359	-0.00368	0.003305	-0.0004	-0.00378	0.003086	0.009283	0.017812	0.0039	-0.00225	-0.02161	0.002207
15	0.00139	-0.01401	0.018094	0.001728	-0.00386	-0.00248	0.002562	0.000363	0.001075	0.009955	0.003348	0.003522	-0.05571	0.001895	-0.00181	-0.00239	-0.02312	-0.03433
16	-0.00201	-0.00119	0.018884	0.000209	-0.00489	-0.00588	-0.00167	0.011111	-0.00096	0.001176	0.000841	0.001544	-0.00962	0.00913	-0.00505	0.002307	0.018405	-0.01554
17	-0.00306	0.013885	0.003977	-0.00089	-0.00399	-0.00179	-0.0058	-0.00726	-0.00132	0.004625	0.000491	0.002326	-0.03709	0.012016	0.002778	-0.0015	0.007361	0.014601
18	-0.00398	-0.01862	0.003094	-0.00286	-0.00461	-0.00801	-0.00447	-0.01059	-0.00036	-0.01999	-0.00159	0.000854	-0.00848	0.010541	-0.02129	-0.00118	-0.01103	-0.00331
19	0.000445	0.002568	-0.02358	-0.00266	-9.5E-05	0.008447	0.000897	-0.00516	0.00281	0.048139	-0.00217	0.00316	-0.00567	-0.00378	-0.00449	-0.00338	-0.01721	0.00997
20	0.003979	-0.01792	-0.00032	-0.00154	0.009903	0.000975	0.00412	-0.00407	-0.00249	0.012279	-0.00272	0.005495	0.001063	0.007653	-0.00275	-0.00199	-0.02758	0.004105
21	0.000832	0.005763	0.01722	-0.00063	0.001935	0.004146	0.000717	0.006089	0.002321	0.030866	0.004675	-0.00472	-0.01869	-0.00036	0.00127	0.001343	0.003816	0.006329
22	0.004981	0.000241	-0.03649	-0.00054	-0.00141	0.000746	0.002733	-0.00839	-0.00056	-0.03336	-0.00301	0.006385	0.016758	0.004781	-0.01818	-0.00283	0.009108	0.029249
23	0.000147	-0.00098	0.029687	0.001286	-0.00014	-0.00206	-0.01203	-0.00691	2.56E-06	-0.0035	-0.003	-0.00265	0.008366	-0.00568	-0.00244	-0.00039	0.020698	-0.01429
24	-0.0021	-0.0089	-0.02246	-0.00039	-0.00801	-0.00555	-0.00098	0.013586	-0.00215	-0.01729	-0.00679	-0.01229	-0.02574	-0.00857	-0.00332	-0.00475	-0.00714	-0.00385
25	0.000888	0.016804	0.039657	-0.00126	0.004001	-0.00376	0.002751	0.001184	-0.0015	-0.04713	-0.00281	-0.00053	0.004004	0.000341	-0.00453	-0.00302	0.023454	0.016756

Figure 4.10: The Matrix U.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	0.565968	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0.360747	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0.338734	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0.283125	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.272349	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0.260873	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0.244294	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0.230147	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0.219114	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0.215103	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0.213072	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0.206364	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0.199811	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.199718	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.195998	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.193893	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.191063	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.187905
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.11: The Matrix ∑.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	-0.06778	0.000626	-0.00271	0.133759	-0.02298	0.027283	-0.0262	0.031649	-0.03736	0.000687	-0.07659	0.000365	-0.0162	0.037071	-0.02479	0.020192	-0.01318	0.005611
2	-0.04969	0.018913	-0.00185	0.084949	0.013864	0.025292	0.007975	0.038092	0.004445	0.002514	0.089837	0.033642	0.028008	-0.10694	0.009057	-0.00854	0.047474	0.064215
3	-0.0497	0.035946	-0.01383	0.062529	0.010234	0.055936	-0.01131	0.07853	-0.04098	0.05069	-0.05841	-0.09477	0.019618	-0.02982	-0.02714	0.02059	0.029948	0.063892
4	-0.06859	0.015914	0.013849	0.107637	0.018794	0.009777	0.000222	-0.00979	-0.0102	0.027881	-0.01779	0.052974	-0.06177	-0.00561	0.021022	0.001749	0.003369	-0.02402
5	-0.05901	0.008108	-0.00025	0.154319	-0.00553	0.030642	-0.00289	0.080664	-0.12898	-0.0278	-0.08692	-0.05691	0.028155	0.06842	-0.08	0.057552	-0.06335	-0.01634
6	-0.04276	0.02712	-0.01236	0.089826	0.030796	0.011566	0.001809	0.016922	0.035198	0.060226	0.131676	0.184251	-0.13725	-0.02326	0.043356	-0.06885	-0.01478	-0.07209
7	-0.07818	0.027588	0.003046	0.129041	0.026876	0.030959	0.013696	0.011479	0.032687	0.061367	0.094743	0.123256	-0.07232	-0.10131	0.052938	0.015645	0.045707	0.003456
8	-0.06325	0.019797	0.019583	0.027877	0.041809	-0.0312	-0.00249	0.005413	0.002231	-0.05286	0.085044	0.072469	-0.02064	-0.03842	-0.0068	-0.06419	-0.07157	-0.00255
9	-0.04894	0.01452	0.011944	0.061802	-0.0012	-0.01902	0.00078	-0.01361	0.060449	0.039388	0.02846	0.057723	0.060263	-0.00856	-0.04801	0.041411	-0.11652	-0.03467
10	-0.04993	0.043631	-0.00484	0.128949	0.023844	0.103747	-0.002	0.147511	-0.08494	-0.03173	0.033472	-0.12014	0.136078	0.040099	-0.04747	-0.00876	0.064727	0.022077
11	-0.03257	-0.00141	-0.00302	0.116338	-0.03367	0.014905	0.034876	0.054863	-0.08791	-0.04028	-0.03285	-0.00028	-0.01447	0.069399	0.005479	0.078378	0.019594	-0.04887
12	-0.06661	0.032922	-0.00739	0.076677	0.041037	0.025608	0.026395	0.034909	0.049647	0.044122	0.12415	0.027326	-0.00172	-0.00053	-0.0198	-0.05186	0.060093	0.008766
13	-0.05182	0.002048	0.025255	0.17784	-0.01331	0.024416	-0.02396	0.032992	-0.11641	-0.01116	-0.08684	-0.00513	0.04449	0.03702	0.002798	0.05068	0.026679	-0.02148
14	-0.06261	0.031517	-0.00906	0.018067	0.005786	-0.00244	-0.00468	-0.00334	-0.00859	0.035488	-0.01326	-0.03551	-0.02514	0.021938	-0.03507	0.026678	0.002399	-0.00245
15	-0.04549	0.027007	-0.00876	0.00756	0.014496	0.01061	0.019257	0.009171	-0.01235	0.003812	-0.01986	-0.08206	-0.03269	0.01163	0.05193	-0.00912	0.075029	0.03396
16	-0.03938	-0.00909	0.046056	0.111782	0.023785	-0.0104	0.004922	0.007673	0.029187	-0.00732	0.098758	0.132131	-0.01055	-0.03534	0.031054	-0.07262	0.023401	-0.02359
17	-0.03217	0.014979	0.001953	0.117256	-0.00379	0.054665	-0.0016	0.098668	-0.10602	-0.08417	-0.08099	-0.07203	0.050846	0.073541	-0.02122	0.046626	-0.02867	-0.01853
18	-0.06428	0.005573	0.013299	0.103837	0.001403	0.000261	-0.00293	-0.04353	0.05474	-0.09425	0.09034	0.155851	0.037034	-0.05056	0.041722	0.027943	0.032353	-0.04302
19	-0.07036	0.057647	-0.03213	0.087412	0.036375	0.025602	-0.00757	0.052796	0.012393	-0.04821	0.02321	0.023945	-0.03069	0.056645	-0.03374	-0.00661	-0.06422	-0.07599
20	-0.04031	0.012827	0.014142	0.022375	-0.02683	-0.02114	-0.01056	-0.04422	0.031105	-0.1151	0.049978	-0.00677	0.07161	0.043006	0.017626	0.010763	-0.00805	-0.02165
21	-0.05316	-0.04085	-0.03363	0.065913	-0.00749	0.006281	-0.07661	-0.01397	0.00126	-0.0324	-0.03551	-0.04029	0.019477	0.072583	-0.01857	-0.02979	-0.00281	-0.01485
22	-0.06706	0.034194	-0.00695	0.069797	0.027906	0.008211	0.015197	0.008514	0.054216	0.050302	0.07136	0.107771	-0.09453	-0.02186	0.022723	-0.01212	0.003378	-0.0485
23	-0.05767	0.002702	0.007631	0.085962	0.030234	0.010839	0.042956	0.001071	0.02258	0.029622	0.170967	0.142658	-0.1148	0.019939	0.05406	-0.06698	0.039503	-0.04653
24	-0.05353	0.00173	0.024815	0.145856	0.021618	0.014543	-0.00304	0.093708	-0.00148	0.00624	0.072732	0.081621	0.074044	-0.12246	0.009602	0.013856	0.03925	0.082111
25	-0.05686	0.04618	-0.01664	0.048537	0.014796	0.064894	-0.00251	0.089476	-0.01815	0.040952	-0.02113	-0.07745	0.017989	-0.07427	-0.0077	-0.01625	0.030101	0.080048

Figure 4.12: The Matrix V^T.

In this stage compute the D matrix according Equation 2.8. Figure 4.13 represent the D matrix.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	-0.03836	0.000226	-0.00092	0.037871	-0.00626	0.007117	-0.0064	0.007284	-0.00819	0.000148	-0.01632	7.54E-05	-0.00324	0.007404	-0.00486	0.003915	-0.00252	0.001054
2	-0.02813	0.006823	-0.00063	0.024051	0.003776	0.006598	0.001948	0.008767	0.000974	0.000541	0.019142	0.006943	0.005596	-0.02136	0.001775	-0.00166	0.00907	0.012066
3	-0.02813	0.012967	-0.00468	0.017703	0.002787	0.014592	-0.00276	0.018074	-0.00898	0.010904	-0.01244	-0.01956	0.00392	-0.00596	-0.00532	0.003992	0.005722	0.012006
4	-0.03882	0.005741	0.004691	0.030475	0.005119	0.002551	5.42E-05	-0.00225	-0.00224	0.005997	-0.00379	0.010932	-0.01234	-0.00112	0.00412	0.000339	0.000644	-0.00451
5	-0.0334	0.002925	-8.5E-05	0.043692	-0.00151	0.007994	-0.00071	0.018565	-0.02826	-0.00598	-0.01852	-0.01174	0.005626	0.013665	-0.01568	0.011159	-0.0121	-0.00307
6	-0.0242	0.009783	-0.00419	0.025432	0.008387	0.003017	0.000442	0.003895	0.007712	0.012955	0.028057	0.038023	-0.02742	-0.00465	0.008498	-0.01335	-0.00282	-0.01355
7	-0.04425	0.009952	0.001032	0.036535	0.00732	0.008076	0.003346	0.002642	0.007162	0.0132	0.020187	0.025436	-0.01445	-0.02023	0.010376	0.003034	0.008733	0.000649
8	-0.0358	0.007142	0.006633	0.007893	0.011387	-0.00814	-0.00061	0.001246	0.000489	-0.01137	0.01812	0.014955	-0.00412	-0.00767	-0.00133	-0.01245	-0.01367	-0.00048
9	-0.0277	0.005238	0.004046	0.017498	-0.00033	-0.00496	0.000191	-0.00313	0.013245	0.008473	0.006064	0.011912	0.012041	-0.00171	-0.00941	0.008029	-0.02226	-0.00651
10	-0.02826	0.01574	-0.00164	0.036509	0.006494	0.027065	-0.00049	0.033949	-0.01861	-0.00683	0.007132	-0.02479	0.02719	0.008008	-0.0093	-0.0017	0.012367	0.004148
11	-0.01843	-0.00051	-0.00102	0.032938	-0.00917	0.003888	0.00852	0.012626	-0.01926	-0.00866	-0.007	-5.7E-05	-0.00289	0.01386	0.001074	0.015197	0.003744	-0.00918
12	-0.0377	0.011877	-0.0025	0.021709	0.011176	0.00668	0.006448	0.008034	0.010878	0.009491	0.026453	0.005639	-0.00034	-0.00011	-0.00388	-0.01006	0.011482	0.001647
13	-0.02933	0.000739	0.008555	0.050351	-0.00363	0.006369	-0.00585	0.007593	-0.02551	-0.0024	-0.0185	-0.00106	0.00889	0.007394	0.000548	0.009827	0.005097	-0.00404
14	-0.03544	0.01137	-0.00307	0.005115	0.001576	-0.00064	-0.00114	-0.00077	-0.00188	0.007634	-0.00283	-0.00733	-0.00502	0.004381	-0.00687	0.005173	0.000458	-0.00046
15	-0.02574	0.009743	-0.00297	0.002141	0.003948	0.002768	0.004704	0.002111	-0.00271	0.00082	-0.00423	-0.01693	-0.00653	0.002323	0.010178	-0.00177	0.014335	0.006381
16	-0.02229	-0.00328	0.015601	0.031648	0.006478	-0.00271	0.001202	0.001766	0.006395	-0.00157	0.021043	0.027267	-0.00211	-0.00706	0.006086	-0.01408	0.004471	-0.00443
17	-0.01821	0.005404	0.000661	0.033198	-0.00103	0.014261	-0.00039	0.022708	-0.02323	-0.0181	-0.01726	-0.01486	0.01016	0.014687	-0.00416	0.00904	-0.00548	-0.00348
18	-0.03638	0.00201	0.004505	0.029399	0.000382	6.81E-05	-0.00072	-0.01002	0.011994	-0.02027	0.019249	0.032162	0.0074	-0.0101	0.008177	0.005418	0.006182	-0.00808
19	-0.03982	0.020796	-0.01089	0.024748	0.009907	0.006679	-0.00185	0.012151	0.002715	-0.01037	0.004945	0.004941	-0.00613	0.011313	-0.00661	-0.00128	-0.01227	-0.01428
20	-0.02281	0.004627	0.00479	0.006335	-0.00731	-0.00551	-0.00258	-0.01018	0.006815	-0.02476	0.010649	-0.0014	0.014309	0.008589	0.003455	0.002087	-0.00154	-0.00407
21	-0.03009	-0.01474	-0.01139	0.018662	-0.00204	0.001639	-0.01872	-0.00322	0.000276	-0.00697	-0.00757	-0.00832	0.003892	0.014496	-0.00364	-0.00578	-0.00054	-0.00279
22	-0.03795	0.012335	-0.00235	0.019761	0.0076	0.002142	0.003712	0.00196	0.011879	0.01082	0.015205	0.02224	-0.01889	-0.00437	0.004454	-0.00235	0.000645	-0.00911
23	-0.03264	0.000975	0.002585	0.024338	0.008234	0.002828	0.010494	0.000247	0.004948	0.006372	0.036428	0.029439	-0.02294	0.003982	0.010596	-0.01299	0.007548	-0.00874
24	-0.0303	0.000624	0.008406	0.041296	0.005888	0.003794	-0.00074	0.021566	-0.00032	0.001342	0.015497	0.016844	0.014795	-0.02446	0.001882	0.002687	0.007499	0.015429
25	-0.03218	0.016659	-0.00564	0.013742	0.00403	0.016929	-0.00061	0.020593	-0.00398	0.008809	-0.0045	-0.01598	0.003594	-0.01483	-0.00151	-0.00315	0.005751	0.015041

Figure 4.13: The Matrix D.

4- Now will be implemented the ICA algorithm to find the sources (ICs) which represent the clusters, where choosing the five vectors with large eigenvalues from the D matrix because it is representing the data which have more informative. This example implemented the FastICA algorithm and running the Argmax function to compute the probability of belonging the text to the one from the five clusters. Figure 4.14 represent the actual labels of clusters texts and predicted values of clusters.

```
Actual cluster   =
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
predicted cluster=
 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3,
```

Figure 4.14: The Actual and Predicted Values of Clusters.

The Figure 4.15 shown the matching matrix, accuracy, and elapsed time.

cluster	C1	C2	C3	C4	C5
IC1	45	4	1	0	5
IC2	0	40	1	0	2
IC3	3	5	47	0	6
IC4	0	1	0	49	0
IC5	2	0	1	1	37

precision	recall	fscore
0.818	0.9	0.859
0.93	0.8	0.865
0.77	0.94	0.855
0.98	0.98	0.98
0.902	0.74	0.821

OverallAccuracy (F-Measure) = 0.872

Elapsed time= 0:00:22.713640

Figure 4.15: Matching Matrix, Accuracy, Elapsed time of Results

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

This dissertation presented two proposed ICA algorithms for unsupervised learning and text clustering. In addition to FastICA algorithm to compare the performance. These algorithms have been proposed to alleviate inherent problems of the traditional ICA algorithm. The suggested algorithm comprises a sequence of processes consisting of centering and whitening, estimating separating (demixing) matrix, restoring sources. Several experiments are conducted and the results produced several conclusions as follows:

1. Since FastICA algorithm is an iterative algorithm that starts with random initialization, depending on the data it can run into local minima. Therefore, used Metaheuristic algorithms to address this drawback because these algorithms can escape from the local minimum and obtain the global solution.
2. Two Metaheuristic optimization algorithms PSO, and GSO were proposed to enhance the performance of the ICA method. These algorithms gave good results better than the standard FastICA method, according to some metrics as precision, recall, f-measure, and overall accuracy.
3. Proving that two proposed PSO-ICA and GSO-ICA algorithms are indeed appropriate methods for text mining applications.
4. ICA clustering methodology presents a good advantage by using the statistical method instead of similarity methods for building a text clustering system.

5. Negentropy function was used for GSO-ICA algorithm which implemented the best results compared with the PSO-ICA algorithm, due to the ability to split agents into subgroups and find multiple global solutions simultaneously.

5.2 Suggestions for Future Work

There are several suggestions that can be handled in the future as follows :

1- Using the proposed algorithms with other text mining applications such as identifying spam E-mails, classification fake tweets, and sentiment analysis.

2- Using other Metaheuristic optimization algorithms such as Genetic Algorithm to improve the traditional ICA algorithm for text mining applications.

3- Employing another functions such as entropy function as an objective function with the proposed algorithm.

4- This application would give new importance to SVD as a method that may become valuable for providing input to ICA in other fields.

REFERENCES

REFERENCES

[1] J. Ma and Z. Sun, "Copula component analysis," in *International Conference on Independent Component Analysis and Signal Separation*, 2007, pp. 73-80.

[2] M. Journée, P.-A. Absil, and R. Sepulchre, "Optimization on the orthogonal group for independent component analysis," in *International Conference on Independent Component Analysis and Signal Separation*, 2007, pp. 57-64.

[3] P. Comon, "Independent component analysis, a new concept?," *Signal processing*, vol. 36, pp. 287-314, 1994.

[4] M. E. Davies, C. C. James, S. A. Abdallah, and M. D. Plumbley, *Independent Component Analysis and Signal Separation: 7th International Conference, ICA 2007, London, UK, September 9-12, 2007, Proceedings* vol. 4666: Springer, 2007.

[5] S. Haykin, "Unsupervised Adaptive Filtering, Volume 1," *Blind source separation*, vol. 1, 2000.

[6] A. Hyvarinen, J. Karhunen, and E. Oja, "Independent component analysis and blind source separation," ed: John Wiley & Sons, 2001.

[7] A. Cichocki and S.-i. Amari, *Adaptive blind signal and image processing: learning algorithms and applications*: John Wiley & Sons, 2002.

[8] F. Wang, H. Li, and R. Li, "Data mining with independent component analysis," in *2006 6th World Congress on Intelligent Control and Automation*, 2006, pp. 6043-6047.

[9] A. Mostafa, "Review of data mining concept and its techniques," *Innovative Technology*, vol. 9, 2016.

[10] R. Talib, M. K. Hanif, S. Ayesha, and F. Fatima, "Text mining: techniques, applications and issues," *International Journal of Advanced Computer Science and Applications*, vol. 7, pp. 414-418, 2016.

[11] W. He, "Examining students' online interaction in a live video streaming environment using data mining and text mining," *Computers in Human Behavior*, vol. 29, pp. 90-102, 2013.

[12] C. Isbell and P. Viola, "Restructuring sparse high dimensional data for effective retrieval," 1998.

[13] A. Kabán, "Unsupervised topic separation and keyword identification in document collections: a projection approach," *http://cis. paisley. ac. uk/research/reports/index. html*, 2000.

[14] X. Sevillano Domínguez, G. Cobo Rodríguez, F. Alías Pujol, and J. C. Socoró Carrié, "Robust document clustering by exploiting feature diversity in cluster ensembles," *Procesamiento del lenguaje natural, nº 37 (sept. 2006), pp. 169-176*, 2006.

[15] A. Chagnaa, C.-Y. Ock, C.-B. Lee, and P. Jaimai, "Feature extraction of concepts by independent component analysis," *Journal of Information Processing Systems*, vol. 3, pp. 33-37, 2007.

[16] X. Yu, D. Hu, and J. Xu, *Blind source separation: theory and applications*: John Wiley & Sons, 2013.

[17] A. Hyvärinen, "Survey on independent component analysis," 1999.

[18] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, pp. 411-430, 2000.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, pp. 1942-1948.

[20] K. N. Kaipa and D. Ghose, *Glowworm swarm optimization: theory, algorithms, and applications* vol. 698: Springer, 2017.

[21] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: classification evaluation," *Nature methods*, vol. 13, pp. 603-604, 2016.

[22] T. Wang, D.-X. Liu, and X.-Z. Lin, "XML document clustering by independent component analysis," in *International Workshop on Knowledge Discovery from XML Documents*, 2006, pp. 13-21.

[23] A. Huang, D. Milne, E. Frank, and I. H. Witten, "Clustering documents using a wikipedia-based concept representation," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009, pp. 628-636.

[24] Q. Pu and D. He, "Semantic clustering based relevance language model," *Information Technology Journal*, vol. 9, pp. 236-246, 2009.

[25] T. Onoda, M. Sakai, and S. Yamada, "Careful seeding method based on independent components analysis for k-means clustering," *Journal of Emerging Technologies in Web Intelligence*, vol. 4, pp. 51-59, 2012.

[26] E. Gultepe and M. Makrehchi, "Improving clustering performance using independent component analysis and unsupervised feature learning," *Human-centric Computing and Information Sciences*, vol. 8, pp. 1-19, 2018.

[27] A. Ghazdali, A. Metrane, and A. Ourdou, "Blind Source Separation for Text Mining," in *Journal of Physics: Conference Series*, 2021, p. 012018.

[28] E. Hasanzadeh and H. A. Rokny, "Text clustering on latent semantic indexing with particle swarm optimization (PSO) algorithm," *International Journal of Physical Sciences*, vol. 7, pp. 16-120, 2012.

[29] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Open Computer Science*, vol. 3, pp. 69-90, 2013.

[30] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *Journal of Computational Science*, vol. 25, pp. 456-466, 2018.

[31] R. Janani and S. Vijayarani, "Text document clustering using spectral clustering algorithm with particle swarm optimization," *Expert Systems with Applications*, vol. 134, pp. 192-200, 2019.

[32] S. Selvaraj and E. Choi, "Swarm Intelligence Algorithms in Text Document Clustering with Various Benchmarks," *Sensors*, vol. 21, p. 3196, 2021.

[33] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*: Cambridge university press, 2007.

[34] G. Salton, "Automatic text processing: The transformation, analysis, and retrieval of," *Reading: Addison-Wesley*, vol. 169, 1989.

[35] P. Tan, M. Steinbach, and V. Kumar, "Introduction to data mining, addison wesley publishers," 2006.

[36] V. Korde and C. N. Mahender, "Text classification and classifiers: A survey," *International Journal of Artificial Intelligence & Applications,* vol. 3, p. 85, 2012.

[37] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, *et al.*, "A brief survey of text mining: Classification, clustering and extraction techniques," *arXiv preprint arXiv:1707.02919,* 2017.

[38] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A hybrid strategy for krill herd algorithm with harmony search algorithm to improve the data clustering," *Intelligent Decision Technologies,* vol. 12, pp. 3-14, 2018.

[39] T. Wei, Y. Lu, H. Chang, Q. Zhou, and X. Bao, "A semantic approach for text clustering using WordNet and lexical chains," *Expert Systems with Applications,* vol. 42, pp. 2264-2275, 2015.

[40] L. Beumer, "Evaluation of Text Document Clustering using k-Means," The University of Wisconsin-Milwaukee, 2020.

[41] G. J. Kowalski and M. T. Maybury, *Information storage and retrieval systems: theory and implementation* vol. 8: Springer Science & Business Media, 2000.

[42] T. Jo, "Text mining," *Studies in Big Data. Cham: Springer International Publishing,* 2019.

[43] C. D. Manning, "Prabhakar raghavan, and hinrich schutze," *Introduction to information retrieval,* 2008.

[44] J. Singh and V. Gupta, "A systematic review of text stemming techniques," *Artificial Intelligence Review,* vol. 48, pp. 157-217, 2017.

[45] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management,* vol. 24, pp. 513-523, 1988.

[46] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722,* 2014.

[47] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on*

empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.

[48] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, p. 150, 2019.

[49] S. Lappin, "13 Curry Typing, Polymorphism, and Fine-Grained Intensionality," *The Handbook of Contemporary Semantic Theory*, p. 408, 2015.

[50] Y. Ko, "A study of term weighting schemes using class information for text classification," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012, pp. 1029-1030.

[51] A. Ozgur, "Supervised and unsupervised machine learning techniques for text document categorization," *Unpublished Master's Thesis, İstanbul: Boğaziçi University*, 2004.

[52] W. Chen, X. Xie, J. Wang, B. Pradhan, H. Hong, D. T. Bui, *et al.*, "A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility," *Catena*, vol. 151, pp. 147-160, 2017.

[53] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, pp. 1-19, 2017.

[54] R. R. Larson, "Introduction to information retrieval," ed: Wiley Online Library, 2010.

[55] M. Awad and R. Khanna, "Support vector machines for classification," in *Efficient Learning Machines*, ed: Springer, 2015, pp. 39-66.

[56] H. T. Sueno, B. D. Gerardo, and R. P. Medina, "Multi-class document classification using support vector machine (SVM) based on improved Naïve bayes vectorization technique," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, 2020.

[57] J. Oyelade, I. Isewon, O. Oladipupo, O. Emebo, Z. Omogbadegun, O. Aromolaran, *et al.*, "Data Clustering: Algorithms and Its Applications," in *2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, 2019, pp. 71-81.

[58] A. Tharwat, "Independent component analysis: An introduction," *Applied Computing and Informatics*, 2020.

[59] J.-T. Chien, *Source separation and machine learning*: Academic Press, 2018.

[60] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE transactions on Neural Networks*, vol. 10, pp. 626-634, 1999.

[61] J. V. Stone, "Independent component analysis," *A Bradford Book*, 2004.

[62] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, pp. 391-407, 1990.

[63] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*: Cambridge university press, 2020.

[64] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, and M. Jordan, "A linear non-Gaussian acyclic model for causal discovery," *Journal of Machine Learning Research*, vol. 7, 2006.

[65] D. Langlois, S. Chartier, and D. Gosselin, "An introduction to independent component analysis: InfoMax and FastICA algorithms," *Tutorials in Quantitative Methods for Psychology*, vol. 6, pp. 31-38, 2010.

[66] B. A. Pearlmutter and L. C. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Advances in neural information processing systems*, 1997, pp. 613-619.

[67] G. R. Naik and D. K. Kumar, "An overview of independent component analysis and its applications," *Informatica*, vol. 35, 2011.

[68] T.-W. Lee, "Independent component analysis," in *Independent component analysis*, ed: Springer, 1998, pp. 27-66.

[69] E. Bingham and A. Hyvärinen, "A fast fixed-point algorithm for independent component analysis of complex valued signals," *International journal of neural systems*, vol. 10, pp. 1-8, 2000.

[70] N. BURSA and H. TATLIDİL, "Evaluation of Independent Components Analysis from Statistical Perspective and Its Comparison

with Principal Components Analysis," *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi,* vol. 24, pp. 474-486, 2020.

[71] A. Dermoune and T. Wei, "FastICA algorithm: five criteria for the optimal choice of the nonlinearity function," *IEEE transactions on signal processing,* vol. 61, pp. 2078-2087, 2013.

[72] Y. Deville and L. T. Duarte, "An overview of blind source separation methods for linear-quadratic and post-nonlinear mixtures," in *International Conference on Latent Variable Analysis and Signal Separation*, 2015, pp. 155-167.

[73] X.-S. Yang, "Metaheuristic optimization: algorithm analysis and open problems," in *International Symposium on Experimental Algorithms*, 2011, pp. 21-32.

[74] E.-G. Talbi, *Metaheuristics: from design to implementation* vol. 74: John Wiley & Sons, 2009.

[75] X.-S. Yang, *Nature-inspired metaheuristic algorithms*: Luniver press, 2010.

[76] M. Dorigo, "Optimization, learning and natural algorithms [Ph. D. thesis]," *Politecnico di Milano, Italy,* 1992.

[77] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation,* vol. 1, pp. 330-343, 2010.

[78] D. Devikanniga, K. Vetrivel, and N. Badrinath, "Review of meta-heuristic optimization based artificial neural networks and its applications," in *Journal of Physics: Conference Series*, 2019, p. 012074.

[79] K.-L. Du and M. Swamy, "Search and optimization by metaheuristics," *Techniques and Algorithms Inspired by Nature,* 2016.

[80] S. Koziel and A. Bekasiewicz, *Multi-objective design of antennas using surrogate models*: World Scientific, 2016.

[81] K. Y. Lee and J.-B. Park, "Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages," in *2006 IEEE PES Power Systems Conference and Exposition,* 2006, pp. 188-192.

[82] R. Dogra and N. Gupta, "Glowworm Swarm Optimization technique for optimal power flow," *Advance in Electronic and Electric Engineering*, vol. 4, pp. 155-160, 2014.

[83] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, 2014, pp. 2376-2381.

[84] Z. Li and X. Huang, "Glowworm swarm optimization and its application to blind signal separation," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[85] A. Nayyar and R. Singh, "Ant colony optimization—computational swarm intelligence technique," in *2016 3rd International conference on computing for sustainable global development (INDIACom)*, 2016, pp. 1493-1499.

[86] U. Goel, S. Varshney, A. Jain, S. Maheshwari, and A. Shukla, "Three dimensional path planning for UAVs in dynamic environment using glow-worm swarm optimization," *Procedia computer science*, vol. 133, pp. 230-239, 2018.

[87] S. Sarkar, A. Roy, and B. S. Purkayastha, "Application of particle swarm optimization in data clustering: A survey," *International Journal of Computer Applications*, vol. 65, 2013.

[88] K. N. Krishnanand and D. Ghose, "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics," in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, 2005, pp. 84-91.

[89] K. Krishnanand and D. Ghose, "Glowworm swarm optimization for multimodal search spaces," in *Handbook of swarm intelligence*, ed: Springer, 2011, pp. 451-467.

[90] P. Lison, "An introduction to machine learning," *Language Technology Group (LTG)*, vol. 1, pp. 1-35, 2015.

[91] R. Veerabhadrappa, M. Ul Hassan, J. Zhang, and A. Bhatti, "Compatibility evaluation of clustering algorithms for contemporary extracellular neural spike sorting," *Frontiers in systems neuroscience*, vol. 14, p. 34, 2020.

[92] C. D. Manning and P. Raghavan, "utze, Introduction to Information Retrieval," ed: Cambridge, UK: Cambridge Univ. Press, 2008.

[93] H. Cunningham, "Encyclopedia of Language and Linguistics 2nd Edition, chapter Information Extraction, Automatic," ed: Elsevier, 2005.

[94] C. H. Ku, A. Iriberri, and G. Leroy, "Crime information extraction from police and witness narrative reports," in *2008 IEEE Conference on Technologies for Homeland Security*, 2008, pp. 193-198.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل-كلية تكنولوجيا المعلومات
قسم البرمجيات

تحسين تحليل المكون المستقل اعتماداً على خوارزميات الادلة العالية لغرض تجميع النصوص

أطروحة مقدمة

إلى مجلس كلية تكنولوجيا المعلومات - جامعة بابل وهي جزء من متطلبات نيل درجة الدكتوراه فلسفة في تكنولوجيا المعلومات – برمجيات

من قبل

حافظ علي شباط مزعل

إشراف

أ. د نداء عبد المحسن عباس

١٤٤٣ هـ ٢٠٢١ م

المستخلص

يعد تحليل المكون المستقل (ICA) طريقة مستخدمة على نطاق واسع لحل مشاكل الفصل الاعمى للمصادر حيث يفترض أن المصادر مستقلة عن بعضها البعض ويتم استخراجها من خلال تعظيم اللاغاوسية Non-Gaussian كدالة هدف. تشتمل طريقة ICA على مكونين، خوارزمية التحسين ودالة الهدف. تستخدم طرق ICA الاكثر شيوعا وظيفة التدرج كدالة هدف لحل هذه المشكلة. عيب هذه الطريقة هي أنها عرضة للوقوع ضمن النهايات الصغرى.

تمتلك خوارزميات Metaheuristic عددًا من المزايا متمثلة بسهولة التنفيذ وإمكانية كبيرة للهروب من الحلول المثلى المحلية. لذلك تم اقتراح خوارزميتي Metaheuristic هما: Particle Swarm Optimization (PSO)، Glowworm Swarm Optimization (GSO)) لمعالجة الحد الأدنى المحلي من العيب وتحسين أداء خوارزمية ICA.

هذه الطرق المقترحة تم تنفيذها لاقتراح نظام لعنقدة النصوص، بما انه يمكن اعتبار المستندات على أنها مزيج من المفاهيم الكامنة التي تجمع الكلمات، لذلك، تم استخدام تحليل القيمة المفردة Singular Value Decomposition (SVD) في التحليل الدلالي الكامن لتحويل الكلمات والمستندات إلى مساحة أبعاد جديدة تعكس المفاهيم الدلالية. حيث تم استخدام المكونات الرئيسية الناتجة من تحليل SVD كمدخلات إلى خوارزميات FastICA ، PSO-ICA، GSO-ICA لاستخراج المكونات المستقلة (ICs) والتي تم استخدامها كمجموعات (clusters). ولتحديد قدرة ICA على تجميع المستندات، تم تحويل ICs إلى "احتمالات المجموعات" باستخدام دالة Argmax .

لتقييم أداء الطرق المقترحة ، تم استخدام مقاييس التقييم النموذجية التي تتضمن (Precision، Recall، F-Measure، Overall Accuracy). أجريت ثلاث تجارب رئيسية، حيث أجريت التجربة الأولى على مجموعة البيانات الطبية والتجارب الأخرى على مجموعة بيانات بي بي سي الإخبارية. النتائج التي تم الحصول عليها باستخدام مجموعة

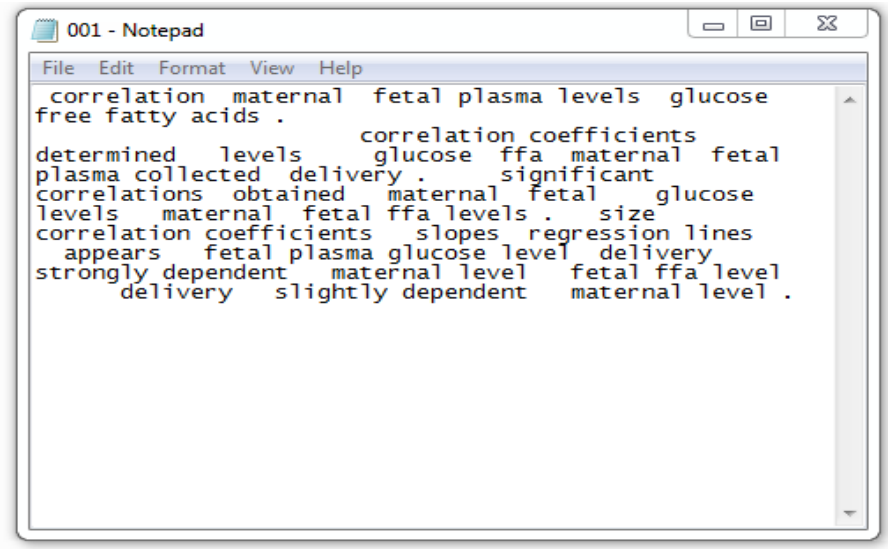
البيانات الطبية كانت ٩٠.٣٪ للدقة الكلية باستخدام خوارزمية GSO-ICA، ٨٩.٥٪ للدقة الكلية باستخدام خوارزمية PSO-ICA، و٨٥.٥٪ للدقة الكلية باستخدام خوارزمية FastICA. النتائج التي تم الحصول عليها باستخدام مجموعة بيانات BBC الإخبارية بلغت ٩١.٦٪ باستخدام خوارزمية GSO-ICA، ٩٠.٨٪ باستخدام خوارزمية PSO-ICA، و ٨٧.٢٪ باستخدام خوارزمية FastICA لمجموعة فرعية من وثائق مجموعة بيانات BBC الإخبارية. أجريت التجربة الأخيرة على مجموعة بيانات BBC الإخبارية بأكملها، وكانت النتائج ٩٠.٣٪ باستخدام خوارزمية GSO-ICA، ٩٠.١٪ باستخدام خوارزمية PSO-ICA و٨٤.٩٪ باستخدام خوارزمية FastICA. توضح هذه النتائج الأداء المتفوق للخوارزميات المقترحة مقارنة بخوارزمية FastICA القياسية التي قدمت اقل وقت تنفيذ بالمقارنة مع الخوارزميات المقترحة. بالإضافة إلى ذلك، أظهرت النتائج التجريبية تفوق GSO-ICA على PSO-ICA نظرًا لقدرتها على تقسيم الوكلاء(agents) إلى مجموعات فرعية وإيجاد حلول عالمية متعددة في وقت واحد.

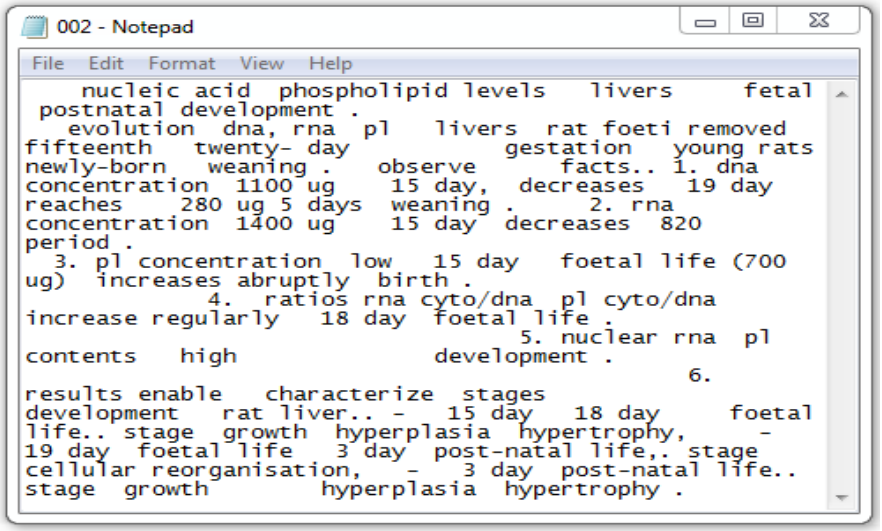
Appendix-A

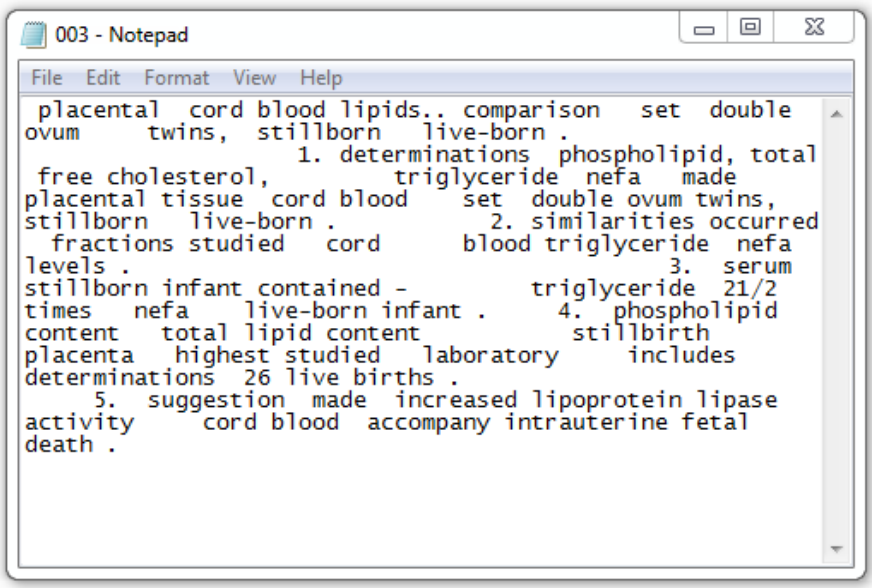
Appendix-A: Samples of datasets

This appendix presents some samples of the datasets that were used to propose a text clustering system. These datasets are MED dataset which contents 124 files and the BBC news dataset which contains 2225 files.

A.1 Samples of MED dataset:



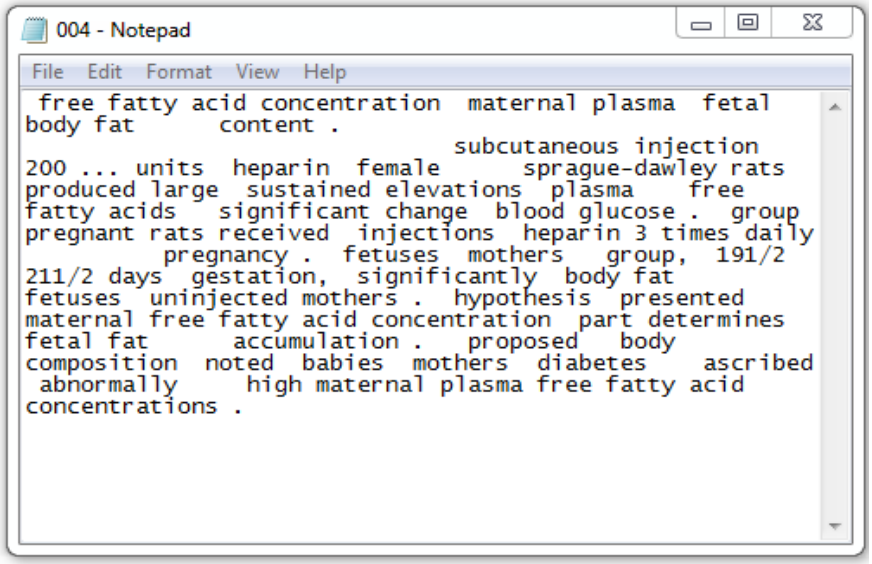




003 - Notepad

File Edit Format View Help

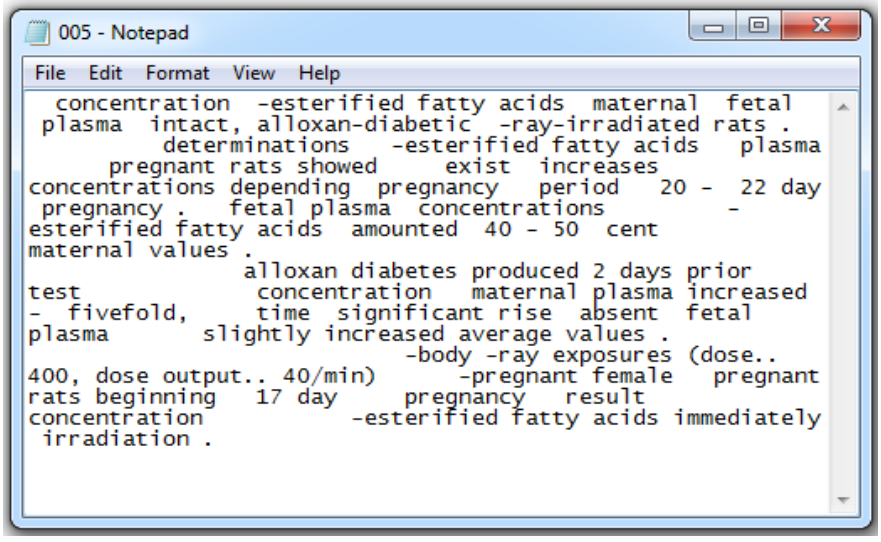
placental cord blood lipids.. comparison set double ovum twins, stillborn live-born .
1. determinations phospholipid, total free cholesterol, triglyceride nefa made placental tissue cord blood set double ovum twins, stillborn live-born . 2. similarities occurred fractions studied cord blood triglyceride nefa levels . 3. serum stillborn infant contained - triglyceride 21/2 times nefa live-born infant . 4. phospholipid content total lipid content stillbirth placenta highest studied laboratory includes determinations 26 live births .
5. suggestion made increased lipoprotein lipase activity cord blood accompany intrauterine fetal death .



004 - Notepad

File Edit Format View Help

free fatty acid concentration maternal plasma fetal body fat content .
subcutaneous injection 200 ... units heparin female sprague-dawley rats produced large sustained elevations plasma free fatty acids significant change blood glucose . group pregnant rats received injections heparin 3 times daily pregnancy . fetuses mothers group, 191/2 211/2 days gestation, significantly body fat fetuses uninjected mothers . hypothesis presented maternal free fatty acid concentration part determines fetal fat accumulation . proposed body composition noted babies mothers diabetes ascribed abnormally high maternal plasma free fatty acid concentrations .

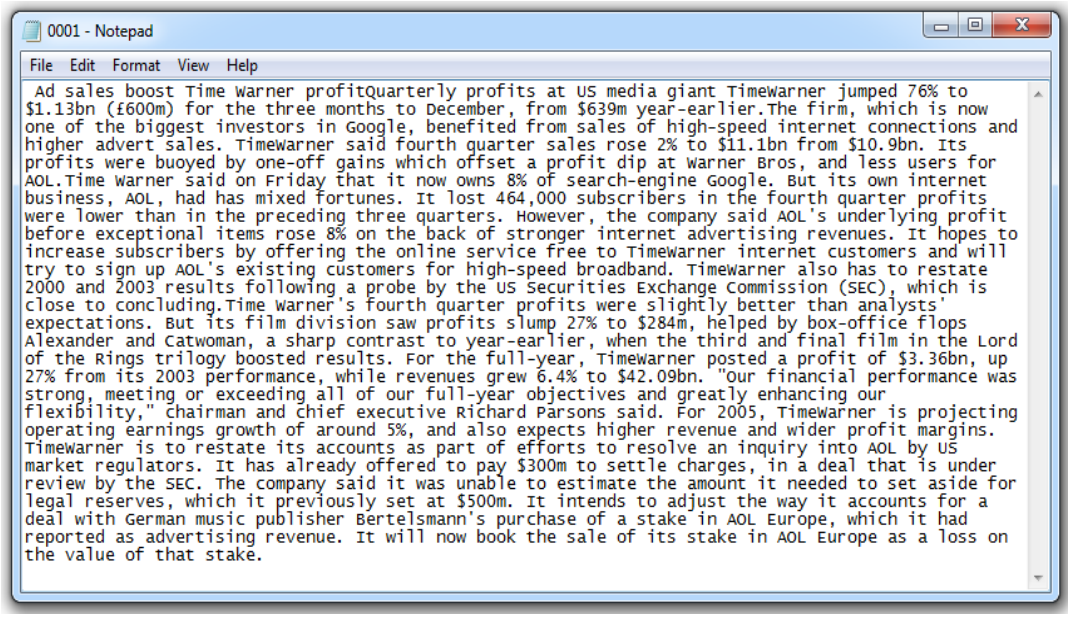


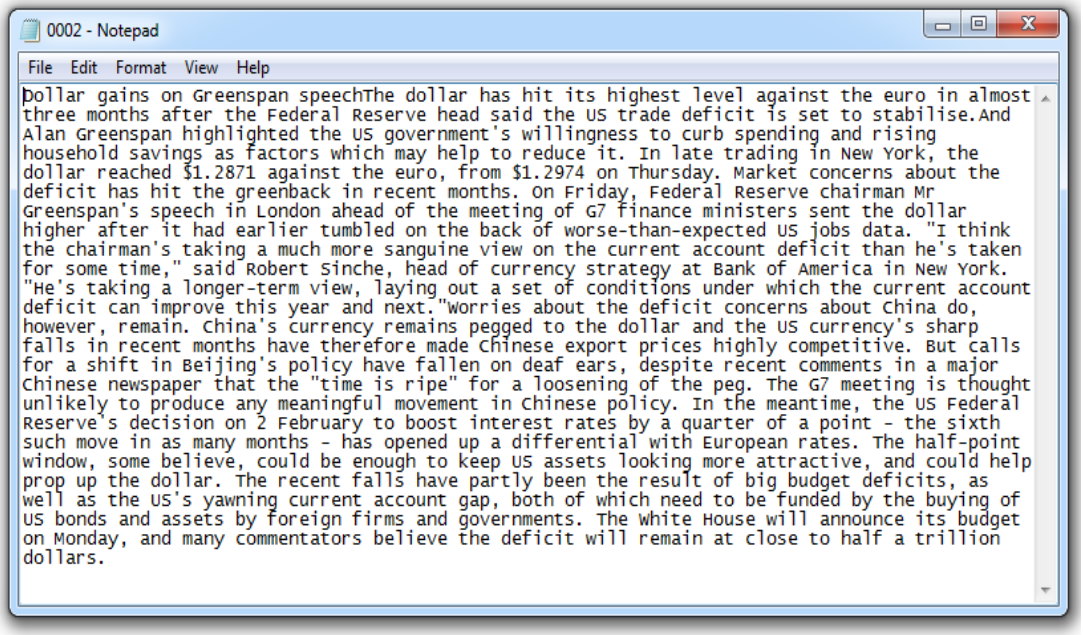
005 - Notepad

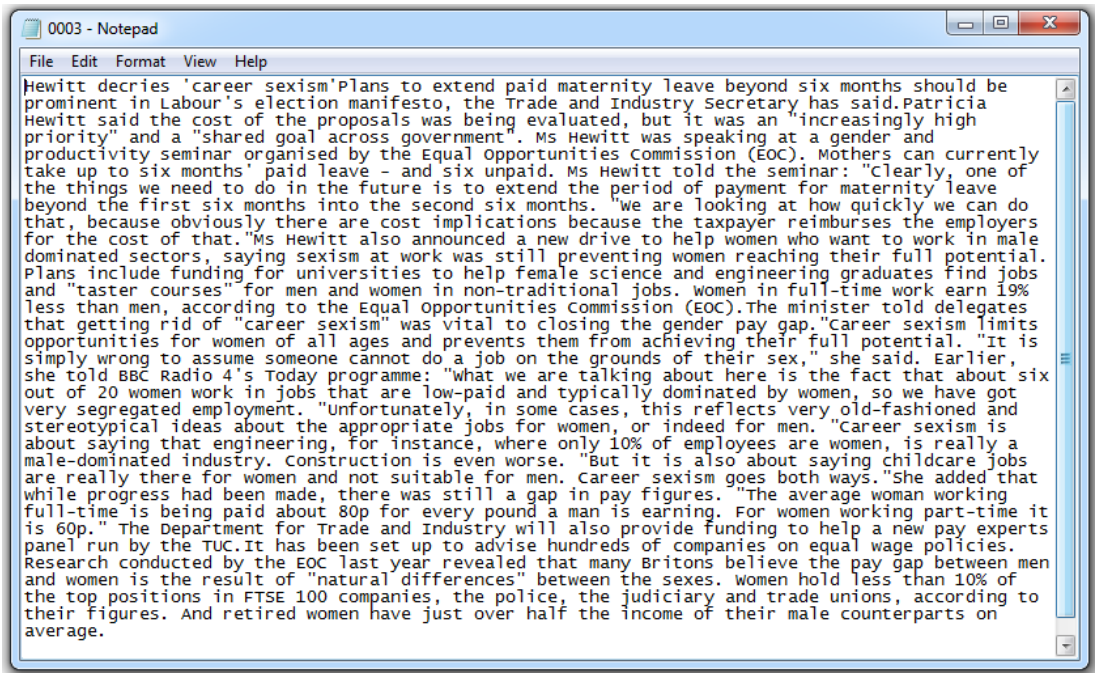
File Edit Format View Help

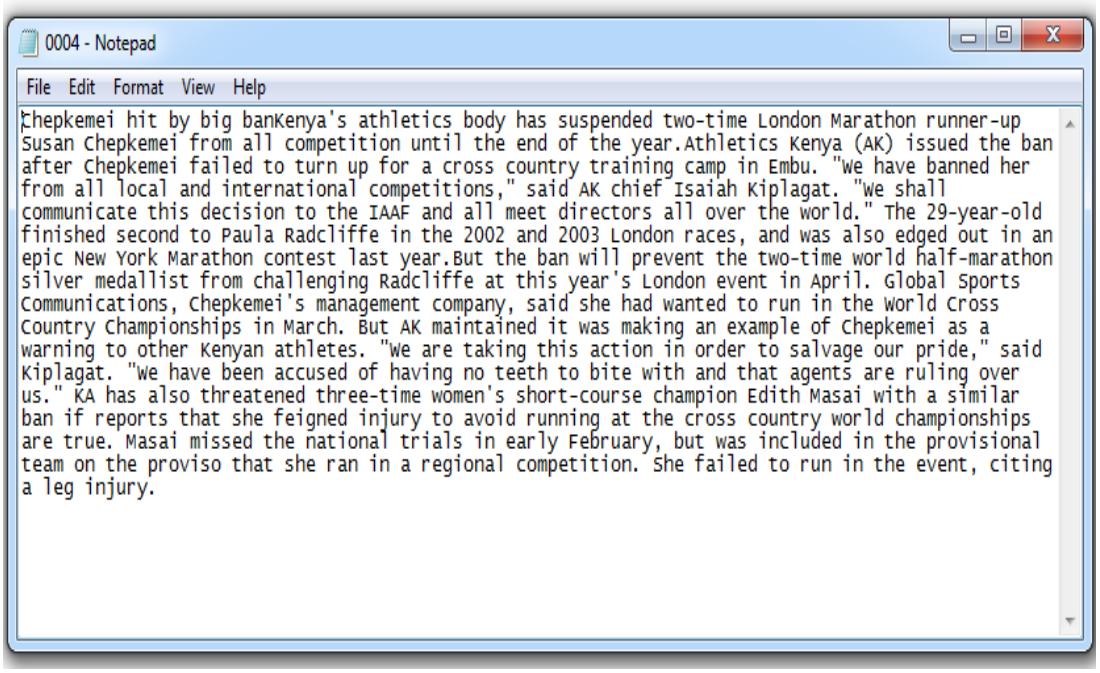
concentration -esterified fatty acids maternal fetal plasma intact, alloxan-diabetic -ray-irradiated rats .
determinations -esterified fatty acids plasma pregnant rats showed exist increases concentrations depending pregnancy period 20 - 22 day pregnancy . fetal plasma concentrations - esterified fatty acids amounted 40 - 50 cent maternal values .
alloxan diabetes produced 2 days prior test concentration maternal plasma increased - fivefold, time significant rise absent fetal plasma slightly increased average values .
-body -ray exposures (dose.. 400, dose output.. 40/min) -pregnant female pregnant rats beginning 17 day pregnancy result concentration -esterified fatty acids immediately irradiation .

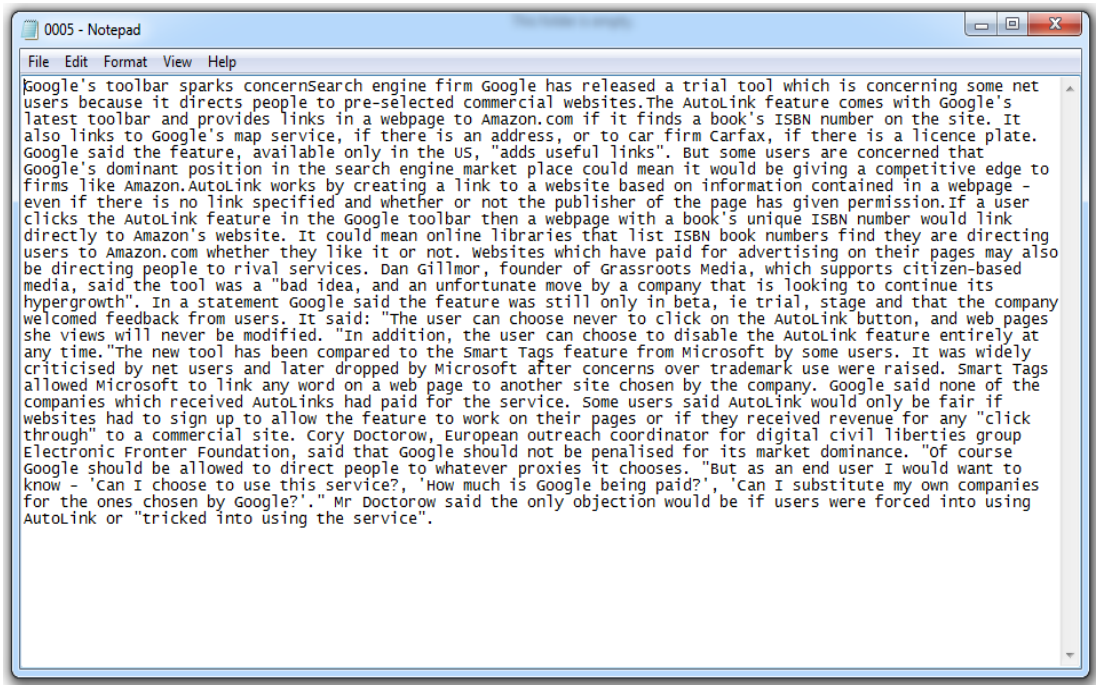
A.2 Samples of BBC news dataset：






0003 - Notepad
File Edit Format View Help
Hewitt decries 'career sexism'Plans to extend paid maternity leave beyond six months should be prominent in Labour's election manifesto, the Trade and Industry Secretary has said.Patricia Hewitt said the cost of the proposals was being evaluated, but it was an "increasingly high priority" and a "shared goal across government". Ms Hewitt was speaking at a gender and productivity seminar organised by the Equal Opportunities Commission (EOC). Mothers can currently take up to six months' paid leave - and six unpaid. Ms Hewitt told the seminar: "Clearly, one of the things we need to do in the future is to extend the period of payment for maternity leave beyond the first six months into the second six months. "We are looking at how quickly we can do that, because obviously there are cost implications because the taxpayer reimburses the employers for the cost of that."Ms Hewitt also announced a new drive to help women who want to work in male dominated sectors, saying sexism at work was still preventing women reaching their full potential. Plans include funding for universities to help female science and engineering graduates find jobs and "taster courses" for men and women in non-traditional jobs. Women in full-time work earn 19% less than men, according to the Equal Opportunities Commission (EOC).The minister told delegates that getting rid of "career sexism" was vital to closing the gender pay gap."Career sexism limits opportunities for women of all ages and prevents them from achieving their full potential. "It is simply wrong to assume someone cannot do a job on the grounds of their sex," she said. Earlier, she told BBC Radio 4's Today programme: "What we are talking about here is the fact that about six out of 20 women work in jobs that are low-paid and typically dominated by women, so we have got very segregated employment. "Unfortunately, in some cases, this reflects very old-fashioned and stereotypical ideas about the appropriate jobs for women, or indeed for men. "Career sexism is about saying that engineering, for instance, where only 10% of employees are women, is really a male-dominated industry. Construction is even worse. "But it is also about saying childcare jobs are really there for women and not suitable for men. Career sexism goes both ways."She added that while progress had been made, there was still a gap in pay figures. "The average woman working full-time is being paid about 80p for every pound a man is earning. For women working part-time it is 60p." The Department for Trade and Industry will also provide funding to help a new pay experts panel run by the TUC.It has been set up to advise hundreds of companies on equal wage policies. Research conducted by the EOC last year revealed that many Britons believe the pay gap between men and women is the result of "natural differences" between the sexes. Women hold less than 10% of the top positions in FTSE 100 companies, the police, the judiciary and trade unions, according to their figures. And retired women have just over half the income of their male counterparts on average.


0004 - Notepad
File Edit Format View Help
Chepkemei hit by big banKenya's athletics body has suspended two-time London Marathon runner-up Susan Chepkemei from all competition until the end of the year.Athletics Kenya (AK) issued the ban after Chepkemei failed to turn up for a cross country training camp in Embu. "We have banned her from all local and international competitions," said AK chief Isaiah Kiplagat. "We shall communicate this decision to the IAAF and all meet directors all over the world." The 29-year-old finished second to Paula Radcliffe in the 2002 and 2003 London races, and was also edged out in an epic New York Marathon contest last year.But the ban will prevent the two-time world half-marathon silver medallist from challenging Radcliffe at this year's London event in April. Global Sports Communications, Chepkemei's management company, said she had wanted to run in the World Cross Country Championships in March. But AK maintained it was making an example of Chepkemei as a warning to other Kenyan athletes. "We are taking this action in order to salvage our pride," said Kiplagat. "We have been accused of having no teeth to bite with and that agents are ruling over us." KA has also threatened three-time women's short-course champion Edith Masai with a similar ban if reports that she feigned injury to avoid running at the cross country world championships are true. Masai missed the national trials in early February, but was included in the provisional team on the proviso that she ran in a regional competition. She failed to run in the event, citing a leg injury.


0005 - Notepad
File Edit Format View Help
Google's toolbar sparks concernSearch engine firm Google has released a trial tool which is concerning some net users because it directs people to pre-selected commercial websites.The AutoLink feature comes with Google's latest toolbar and provides links in a webpage to Amazon.com if it finds a book's ISBN number on the site. It also links to Google's map service, if there is an address, or to car firm Carfax, if there is a licence plate. Google said the feature, available only in the US, "adds useful links". But some users are concerned that Google's dominant position in the search engine market place could mean it would be giving a competitive edge to firms like Amazon.AutoLink works by creating a link to a website based on information contained in a webpage - even if there is no link specified and whether or not the publisher of the page has given permission.If a user clicks the AutoLink feature in the Google toolbar then a webpage with a book's unique ISBN number would link directly to Amazon's website. It could mean online libraries that list ISBN book numbers find they are directing users to Amazon.com whether they like it or not. Websites which have paid for advertising on their pages may also be directing people to rival services. Dan Gillmor, founder of Grassroots Media, which supports citizen-based media, said the tool was a "bad idea, and an unfortunate move by a company that is looking to continue its hypergrowth". In a statement Google said the feature was still only in beta, ie trial, stage and that the company welcomed feedback from users. It said: "The user can choose never to click on the AutoLink button, and web pages she views will never be modified. "In addition, the user can choose to disable the AutoLink feature entirely at any time."The new tool has been compared to the Smart Tags feature from Microsoft by some users. It was widely criticised by net users and later dropped by Microsoft after concerns over trademark use were raised. Smart Tags allowed Microsoft to link any word on a web page to another site chosen by the company. Google said none of the companies which received AutoLinks had paid for the service. Some users said AutoLink would only be fair if websites had to sign up to allow the feature to work on their pages or if they received revenue for any "click through" to a commercial site. Cory Doctorow, European outreach coordinator for digital civil liberties group Electronic Fronter Foundation, said that Google should not be penalised for its market dominance. "Of course Google should be allowed to direct people to whatever proxies it chooses. "But as an end user I would want to know - 'Can I choose to use this service?, 'How much is Google being paid?', 'Can I substitute my own companies for the ones chosen by Google?'." Mr Doctorow said the only objection would be if users were forced into using AutoLink or "tricked into using the service".