نموذج رقم (1)

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Evaluating the Effect of Preprocessing in Arabic Documents Clustering

تقييم تأثير المعالجة المسبقة في عنقدة المستندات العربية

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

DECLARATION

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

اسم الطالب: أسامة عبد الفتاح غانم

التوقيع:

التاريخ: 2014/3/31

Student's name: Osama A. Ghanem

Signature:

Date:31/3/2014

Islamic University, Gaza, Palestine
Research and Graduate Affairs
Faculty of Engineering
Computer Engineering Department

Evaluating the Effect of Preprocessing in Arabic Documents Clustering

Osama Abdel Fattah Ghanem

Supervisor

Dr. Mohammed Alhanjouri

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1435H (2014)

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza
مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي 1150

الرقم Ref ج س غ/35/
2014/03/31
التاريخ Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ أسامة عبدالفتاح رجب غانم لنيل درجة الماجستير في كلية *الهندسة* قسم هندسة الحاسوب وموضوعها:

تقييم تأثير المعاجلة المسبقة في عنقدة المستندات العربية

Evaluating the Effect of Preprocessing in Arabic Documents Clustering

وبعد المناقشة التي تمت اليوم الاثنين 30 جمادى الأولى 1435هـ، الموافق 2014/03/31م الساعة الحادية عشرة صباحًا، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. محمد أحمد الحنجوري	مشرفاً ورئيساً	
أ.د. إبراهيم سليمان أبو هيبة	مناقشاً داخلياً	
د. إيهاب صلاح زقّوت	مناقشاً خارجيًا	31-3-2014

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية *الهندسة* / قسم هندسة الحاسوب.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق ،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 ص.ب. 108 الرمال، غزة، فلسطين هاتف
public@iugaza.edu.ps www.iugaza.edu.ps

Acknowledgements

Praise is to Allah; First and foremost, I wish to thank Allah for giving me strength and courage to complete this research.

I would like to thank my parents who have given credit after God in all things. I also extend my thanks a lot to my wife who has supported me throughout my study.

I would like to express my gratitude to my supervisor Dr. Mohammed Alhanjoury, for providing me the opportunity to develop this work over the master years, for his constant guidance, support and motivation.

Special thanks for my instructor
Prof. Ibrahim Abuhaiba and Dr. Ihab Zakout for their valuable guide and comments.

I must also thank my friend's dr. Said Alzebda and Mohammed Azara who helped me in this research.

Last but not least, I would like to thank my family members for giving me the support and love to complete my work.

Osama Ghanem

March, 2014

Table of Contents

List of Abbreviations

BBC	British Broadcasting Corporation
BOT	Bag of Tokens
CA	Classical Arabic
CCA	Corpus of Contemporary Arabic
CNN	Cable News Network
CPU	Central Processing Unit
DA	Dialectal Arabic
DM	Data Mining
EM	Expectation Maximization
FICH	Frequent Itemset-based Hierarchical Clustering
HAC	Hierarchical Agglomerative Clustering
HKM	Hierarchical K-Means Like clustering
IDF	Inverse Document Frequency
IR	Information Retrieval
KL	Kullback-Leibler
MSA	Modern Standard Arabic
NLP	Natural Language Processing
PDF	Probability density function
SR	Sebawai root extractor
SVM	Support Vector Machine
TC	Text Clustering / Text Categorization
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TM	Text Mining
VSM	Vector Space Model
WEKA	Waikato Environment for Knowledge Analysis

List of Figures

List of Tables

تقييم تأثير المعالجة المسبقة في عنقدة المستندات العربية

أسامة عبد الفتاح غانم

الملخص

عنقدة (تصنيف أو تجميع تلقائي) المستندات والنصوص هي تقنية هامة في عملية استرجاع البيانات والمستندات، فهي تهدف إلى تصنيف المستندات إلى مجموعات ذات مغزى متقاربة مع بعضها البعض في المحتوى. عملية تجهيز المستندات يلعب دورا رئيسيا في تحسين عملية عنقدة المستندات العربية.

يتناول هذا البحث ويقارن تقنيات تجهيز مستندات مكتوبة باللغة العربية وتأثيرها في عملية العنقدة أو التجميع التلقائي.

يدرس البحث تقنيات تجهيز النص قبل عملية العنقدة أو التجميع التلقائي وهي: تقليم الكلمات ، توزين الكلمات ومعالجة النصوص مصرفياً (التجذير والتجذير الخفيف للكلمات العربية).

قمنا بعمل تجارب عملية بتطبيق خوارزميات خاصة بعملية العنقدة أو التجميع التلقائي من خلال استخدام خوارزميات تعتمد على التقسيم في عملية التجميع ، وبشكل أساسي استخدمنا خوارزمية (K-means)، ثم تم مقارنة النتائج بخوازمية أخرى من نفس النوع خوارزمية (EM) لمناقشة مدى فاعلية استخدام هذه الخوارزميات في عملية العنقدة.

تم التحقق من استخدام دوال قياس التشابه في عملية العنقدة من خلال مقارنة الدالة الأساسية المستخدمة في قياس التشابه مع دالة أخرى.

عملية التقييم للنتائج تمت باستخدام معايير تقييم مشهورة في تقييم استرجاع البيانات، وقد أظهرت النتائج أن اختيار التقنيات الخاصة بتجهيز المستندات والنصوص لعملية العنقدة له دور جوهري في تحسين النتائج وفق معايير وإعدادات معينة تتوافق مع التعقيد الموجود في اللغة العربية.

أظهرت النتائج تفوق الخوارزمية (K-means) باستخدام دالة قياس التشابه (Euclidean distance)، واستخدام تقنيات تجهيز النصوص قبل عملية العنقدة حسّن العملية باختيار قيمة صغيرة لتردد الكلمات في عملية تقليم الكلمات، وتطبيق التوزين، بالإضافة لاستخدام والتجذير الخفيف للكلمات العربية.

Evaluating the Effect of Preprocessing in Arabic Documents Clustering

Osama Abdel Fattah Ghanem

ABSTRACT

Clustering of text documents is an important technique for documents retrieval. It aims to organize documents into meaningful groups or clusters. Preprocessing text plays a main role in enhancing clustering process of Arabic documents. This research examines and compares text preprocessing techniques in Arabic document clustering. It also studies effectiveness of text preprocessing techniques: term pruning, term weighting using (TF-IDF), morphological analysis techniques using (root-based stemming, light stemming, and raw text), and normalization. Experimental work examined the effect of clustering algorithms using a most widely used partitional algorithm, K-means, compared with other clustering partitional algorithm, Expectation Maximization (EM) algorithm. Comparison between the effect of both Euclidean Distance and Manhattan similarity measurement function was attempted in order to produce best results in document clustering.

Results were investigated by measuring evaluation of clustered documents in many cases of preprocessing techniques. The most frequent and basic measures for text mining evaluation, precision and recall, were used for evaluation measurements. In addition to F-Measure, which used as a combination of precision and recall.

Experimental results show that evaluation of document clustering can be enhanced by implementing term weighting (TF-IDF) and term pruning with small value for minimum term frequency. In morphological analysis, light stemming, is found more appropriate than root-based stemming and raw text. Normalization, also improved clustering process of Arabic documents, and evaluation is enhanced. Finally, K-means in document clustering was found more efficient than EM algorithm, and Euclidean distance similarity measurement function is superior.

Keywords: Arabic Text Mining, Arabic document clustering, Arabic text preprocessing, Term weighting, Arabic morphological analysis (Arabic stemming / light stemming), Vector Space Mode (VSM), TF-IDF, K-means, EM.

CHAPTER 1: INTRODUCTION

The amount of electronic text available, such as electronic publications, electronic books, news articles and web pages is increasing rapidly. As the volume of online text information increases, the challenge of extracting relevant knowledge increases as well. The need for tools that help people to find, filter and manage these resources has grown. Thus, automatic organization of text document collections has become an important research issue. A number of machine learning techniques have been proposed to enhance automatic organization of text data. These techniques can be grouped in two main categories, supervised (document classification) and unsupervised (document clustering) [1].

This chapter introduces text mining, document clustering, describes Arabic language, and investigates Arabic language complexity, finally states motivation, problem and objectives of research.

1.1 Text Mining

Text mining is used to describe the application of data mining techniques to automated discovery of useful or interesting knowledge from unstructured text [2]. Various text mining tasks can be performed on the extracted keywords, tags or semantic information. These include document clustering, classification, information extraction, association analysis and trend analysis [2].

Figure 1.1 [3] depicts a generic process model for a text mining application. The presented model starts with a collection of documents and a text mining tool to retrieve a particular document and preprocess it by checking format and character sets. Then, it goes through a text analysis phase where specific techniques are repeated until information is extracted. Three text analysis techniques are shown in the example; however, many other

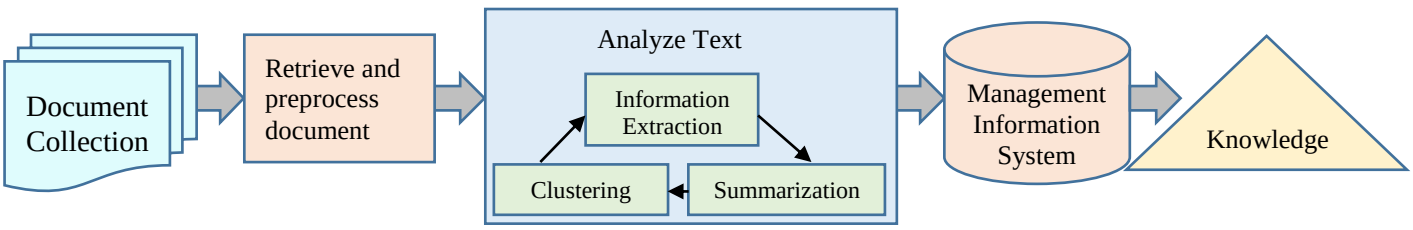

Figure 1.1:An example of text mining

combinations of techniques could be used depending on the goals of the organization. The resulting information can be placed in a management information system, yielding an abundant amount of knowledge for users [3]. The applications of text mining cover a wide range including the following [4]:

1.1.1 Information Retrieval (IR)

Information Retrieval (IR) is defined as: matching a user's query against many unstructured text documents with the purpose of finding the documents that satisfy the user's information needs [5]. Three main approaches are used for matching queries, as follows [6]:

i) Probabilistic retrieval,
ii) Knowledge based IR,
iii) Learning systems based IR.

Probabilistic retrieval is based on estimating a probability of relevance of a certain document to the user's query. On the other hand, a model of the system user and the expert's knowledge is presented in the knowledge based approach. In learning based systems, a machine learning technique is applied in order to extract knowledge and identify patterns in the documents. Learning systems can automatically extract data from examples, and thus, they are more flexible than knowledge based systems. Additionally, unlike probabilistic retrieval systems, they do not suffer from parameters estimation problems [6].

1.1.2 Text Categorization (TC)

Text Categorization (TC) is the process of assigning one or more label to a given text. It is considered as a supervised classification since a collection of labeled (pre–classified) documents is provided. The task is to assign a label to a newly encountered, yet unlabeled, pattern [7]. The most commonly used approach for classification is based on machine learning (ML) techniques [8]. ML is a general inductive process that automatically builds a classifier by learning the characteristics of the categories using a set of pre–classified documents. This is in contrast to the knowledge engineering (KE) based

approach. KE is the process of manually defining a set of rules encoding expert's knowledge on how to classify documents under the given categories. The advantages of ML over KE include considerable savings in terms of expert labor power and straightforward portability to different domains [8].

1.1.3 Text Clustering (TC)

Text Clustering (TC) is considered as an unsupervised learning process. The main aim of TC is to group a collection of unlabeled documents into meaningful clusters that are similar within themselves and dissimilar to documents in other clusters [9]. Clustering documents is attractive because it frees organizations from the need of manually organize document bases, which could be too expensive, or even infeasible given the time constraints of the application and/or the number of documents involved. Machine learning algorithms used for text clustering can be categorized into two main groups, (i) hierarchical clustering algorithms and (ii) partition-based clustering algorithms [10]. Hierarchical clustering algorithms produce nested partitions of data by merging or splitting clusters based on the similarity among them [11]. On the other hand, partition-based clustering algorithms group the data into non–overlapping partitions that usually locally optimize a clustering criterion [12]. Text or document clustering will be discussed in details in chapter 3 as the research scope of text mining.

1.1.4 Text Summarization

Text Summarization is the process of constructing a compressed summary text from the original document according to the user's needs [13]. Summarization is performed using either extraction or abstraction. In extraction, important sentences are extracted from the document and gathered to form document summary. On the other hand, abstraction analyzes the document and provides a better summary using a heavy machinery from natural language processing in addition to some commonsense and domain knowledge data [14].

1.2 Clustering

Clustering is an unsupervised process through which objects are classified into groups called clusters. The problem of clustering is to group unlabeled collection into meaningful clusters without any prior information. Any labels associated with objects are obtained solely from the data. Clustering is useful in a wide range of data analysis fields, including data mining, document retrieval, image segmentation, and pattern classification. In many such problems, little prior information is available about the data, and the decision-maker must make as few assumptions about the data as possible [15]. clustering process can be divided into four stages outlined below [16]:

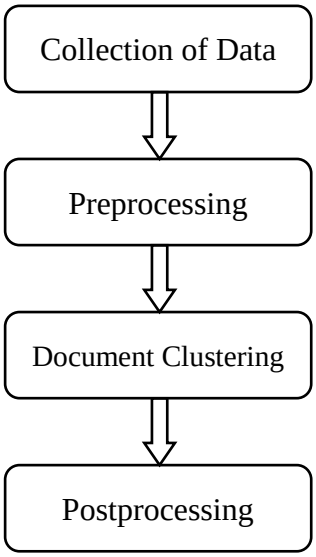

Figure 1.2: The Stages of the Process of Clustering

Collection of Data: includes the processes like crawling, indexing, filtering, etc., which are used to collect documents need to be clustered, index them to store and retrieve in a better way, and filter them to remove the extra data, for example, stop words.

Preprocessing: is done to represent the data in a form that can be used for clustering. There are many ways of representing the documents like, Vector-Model, graphical model, etc. Many measures are also used for weighing the documents and their similarities.

Document Clustering: this topic will be discussed in details in Chapter 3.

Postprocessing: includes the major applications in which the document clustering is used, for example, the recommendation application which uses the results of clustering for recommending news articles to the users.

1.3 Arabic Language

Arabic is one of the 5th widely used languages in the world. It is used by more than 280 million people as the first language, and by 250 million as the second language. Due to the unique nature of Arabic language morphological principles [17], there are relatively few studies on the retrieval/mining of Arabic text documents in the literature.

Arabic language has 3 forms; Classical Arabic (*CA*), Modern Standard Arabic (*MSA*), and Dialectal Arabic (*DA*). *CA, MSA, and DA* forms include classical historical liturgical text, news media and formal speech, and predominantly spoken vernaculars and have no written standards, respectively. Arabic alphabet consists of the following 28 letters

(أ، ب، ت، ث، ج، ح، خ، د، ذ، ر، ز، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ك، ل، م، ن، ه، و، ي)

in addition, the Hamza (ء). Unlike English language, there is no upper or lower case for Arabic letters. The letters (ا،و،ي) are vowels, and the rest are constants. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right to left.

The Arabic script has numerous diacritics, including I'jam (إعجام), consonant pointing, and tashkil (تشكيل), supplementary diacritics. The latter include the ḥarakat (حركات, singular haraka حركة), vowel marks. The literal meaning of tashkil is "forming". As the normal Arabic text does not provide enough information about the correct pronunciation, the main purpose of tashkil (and ḥarakat) is to provide a phonetic guide or a phonetic aid; i.e. show the correct pronunciation (double the word in pronunciation or to act as short vowels). The ḥarakat, which literally means "motions", are the short vowel marks[18]. Arabic diacritics include Fatha, Kasra, Damma, Sukūn, Shadda, and Tanwin. Arabic words have two genders, masculine (مذكر) and feminine

(مؤنث); three numbers, singular (مفرد), dual (مؤنث), and plural (جمع); and three grammatical cases, nominative (الرفع), accusative (النصب), and genitive (الجر). A noun has the nominative case when it is subject (فاعل); accusative when it is the object of a verb (مفعول); and the genitive when it is the object of a preposition (مجرور بحرف جر). Words are classified into three main parts of speech, nouns (أسماء) (including adjectives (صفات) and adverbs (ظروف)), verbs (أفعال), and particles (ادوات) [19].

1.3.1 Arabic Language Challenges

Arabic is a challenging language for a number of reasons [17]:

1. Orthographic with diacritics is less ambiguous and more phonetic in Arabic, certain combinations of characters can be written in different ways. For example, sometimes in glyphs combining HAMZA with ALEF (أ) the HAMZA is dropped (ا). This makes the glyph ambiguous as to whether the HAMZA is present.
2. Arabic has a very complex morphology recording as compared to English language. For example, to convey the possessive, a word shall have the letter (ي) attached to it as a suffix. There is no disjoint Arabic-equivalent of "my".
3. Arabic words are usually derived from a root (a simple bare verb form) that usually contains three letters. In some derivations, one or more of the root letters may be dropped. In such cases tracing the root of the derived word would be a much more difficult problem.
4. Broken plurals are common. Broken plurals are somewhat like irregular English plurals except that they often do not resemble the singular form as closely as irregular plurals resemble the singular in English. Because broken plurals do not obey normal morphological rules, they are not handled by existing stemmers.
5. In Arabic we have short vowels which give different pronunciation. Grammatically they are required but omitted in written Arabic texts.
6. Arabic synonyms are widespread. Arabic is considered as one of the richest languages in the world. This makes exact keyword match is inadequate for Arabic retrieval and classification.

To pass these challenges we will discuss a set of preprocessing routines in chapter 4 to appropriate with clustering process.

1.4 Research Motivation

Electronic documents is increasing rapidly because of amazing progress of computer hardware technology and storage capacities, so machine learning is a powerful solution for automatic categorization of documents and huge data. Document clustering unsupervised learning used widely in other languages in special English language, but fairly limited used in Arabic language which gives great encouragement and motivation to apply clustering process for Arabic language. Classification is a supervised leaning and used more widely in documents categorization, unlike clustering technique which is unsupervised leaning and is limited used for documents despite of importance and efficiency of clustering. Preprocessing text play a central role in enhancement clustering process of Arabic documents, many combinations of preprocessing procedures can be performed; preprocessing impact in clustering Arabic documents is area of research.

1.5 Research Obstacles

- Paucity of implementing clustering for Arabic documents.
- The lack availability of Arabic datasets.
- Large time consumption in experiments because of using huge dataset and clustering process needs much iteration to perform algorithm.
- Huge computer resources needed for performing clustering process in-home using machine-learning tools.

1.6 Research Objectives

1.6.1 Main Objective

The main objective of the research is to cluster Arabic documents using partition-based algorithm, to give best performance for evaluation, by selecting best combinations of text preprocessing, best clustering algorithm, and best similarity measurement function.

1.6.2 Specific Objectives

The specific objectives of the research are:

- Study impact of text preprocessing in clustering evaluation.
- Evaluate clustering process in Arabic document using K-means algorithm, according to recall, precision, F-measure evaluation to build model.
- Study if K-means algorithm is appropriate for Arabic text.
- Use machine learning tool at home for clustering experiments, (WEKA) which is an excellent open-source of data mining tool in abroad, but it is rarely used at home.
- Provide comprehensive guide for using best text preprocessing combination for best clustering evaluation.
- Applying several Arabic morphological analysis tools.

1.7 Research Scope and Limitations

The research has the following Scope and limitations:

1. The research will not modify K-means clustering algorithm.
2. The best results will be compared to other famous clustering algorithms.
3. The experiments of the best obtained results will be applied using other clustering distance measurement method.

1.8 Thesis Organization

The rest of thesis is organized into other 6 chapters as follows:

A detailed study of related work in Arabic text clustering will be presented in chapter 2. In chapter 3 we introduce clustering of documents using famous clustering techniques. System methodology of Arabic document clustering and preprocessing techniques is presented in Chapter 4. Experimental results and analysis of using many combinations of text preprocessing, compared with

other clustering algorithms and distance measurement method are depicted in chapter 5. Finally, chapter 6 concludes our work and suggests future work.

CHAPTER 2: RELATED WORK

2.1 Introduction

This chapter will discuss various works related to this research. Arabic Document Classification was discussed more widely than document clustering which is rarely discussed in Arabic language. In the other hand, many researches have discussed document clustering in English and Chinese and Turkish languages.

In [20], **Singh et al.** applied flat clustering algorithms to documents, in combining with different representation schemes. They concluded that (TF-IDF) representation, and use of stemming obtains better clustering.

Sandhya et al. [21] studied the impact of stemming algorithm along with four similarity measures (Euclidean, cosine, Pearson correlation and extended Jaccard) in conjunction with different types of vector representation (Boolean, term frequency and term frequency, and inverse document frequency) on cluster quality. They concluded that there are four components that affect the results representation of the documents: applying the stemming algorithms, distance or similarity measures considered, and the clustering algorithm itself.

Volkan and Turagay [22] evaluated the impact of stemming on clustering Turkish texts. They conclude that there is no significant evidence that stemming always improves the quality of clustering for texts in Turkish.

Han et al. [23] conducted a Chinese document clustering based on WEKA. They provided a comparison experiment for the improvement of Chinese document clustering. They concluded WEKA is an excellent data mining tool can be used at home which is rarely used at home for document clustering.

However, state of the arts about Arabic document clustering is introduced in this chapter and the researches are fall into three categories: K-means and other algorithms, Text preprocessing, and similarity/distance measures in document clustering.

2.2 K-Means and Other algorithms in document clustering:

Alkoffash [24] implemented the K-means and K-mediods algorithms in order to make a practical comparison between them. The system was tested using a manual set of clusters that consists from 242 predefined clustering documents. The results showed a good indication about using them especially for K-mediods. The average precision and recall for K-means compared with K-mediods are 0.56, 0.52, 0.69 and 0.60 respectively. He also extracted feature set of keywords in order to improve the performance, the result illustrated that two algorithms can be applied to Arabic text, a sufficient number of examples for each category, the selection of the feature space, the training data set used and the value of K can enormously affect the accuracy of clustering. Recall and precision measurers are used for evaluation. Results show K-mediods is better than K-means due to the chance that is given for several files in K--mediods to become a center for a given cluster. Evaluation for K-mediods: 0.60, 0.69 for Average Recall, and Average Precision respectively, despite evaluation for K-means: 0.525, 0.565 for Average Recall, and Average Precision respectively. He concluded that manipulating large corpus may give results that are more nearby to the manual one. Clustering environment is more unbiased than manual due to its dependability on the system rather than user opinion. Most of the errors or weakness that appear in Arabic retrieval systems, due to the strength of language itself that contains several features not existed in any other one. The problem of K-means and K-mediods are represented by selecting initial points, problems of differing sizes, densities, and shapes and outliers data.

Ghwanmeh[25] implemented clustering technique which is K-Means like with hierarchical initial set (Hierarchical K-Means Like clustering HKM). He proved that clustering document sets do enhancement precision on information retrieval systems, since it was proved by Bellot & El-Beze on French language. He made comparison between the traditional information retrieval system and the clustered one. Also the effect of increasing number of clusters on precision is studied. The indexing technique is Term Frequency * Inverse Document Frequency (TF-IDF). It has been found that the effect of

Hierarchical K-Means Like clustering (HKM) with 3 clusters over 242 Arabic abstract documents from the Saudi Arabian National Computer Conference has significant results compared with traditional information retrieval system without clustering. Additionally it has been found that it is not necessary to increase the number of clusters to improve precision more. He applied 59 queries on 242 Arabic abstract documents, which are clustered into several sets of clusters (2, 3 and 5), then he compared the results with the traditional IR system. To determine the appropriate number of clusters; a series of tests have been made at several number of clusters (2, 3, and 5), and was found that the best results is at 3 clusters which means that this corpora talks mainly about three topics. In his results the best precision was obtained is 0.49 which enhances results without using clustering by 13%.

Rafi et al. [26] compared and contrast two approaches to document clustering based on suffix tree data model. The first is an Efficient Phrase based document clustering, which extracts phrases from documents to form compact document representation and uses a similarity measure based on common suffix tree to cluster the documents. The second approach is a frequent word/word meaning sequence based document clustering, it similarly extracts the common word sequence from the document and uses the common sequence/ common word meaning sequence to perform the compact representation, and finally, it uses document clustering approach to cluster the compact documents. These algorithms are using agglomerative hierarchical document clustering to perform the actual clustering step, the difference in these approaches are mainly based on extraction of phrases, model representation as a compact document, and the similarity measures used for clustering. They investigated the computational aspect of the two algorithms, and the quality of results they produced. The result of experiment shows that the F-score obtained from the test data sets clearly exhibits the superiority of algorithm "Efficient Phrase based clustering algorithm" over algorithm "Text document clustering based on frequent word meaning sequences", on variety of situations. They clearly concluded from the results obtained that Efficient Phrase based clustering algorithm is superior.

Al-sarrayrih, and Al-Shalabi[27] used "Frequent Itemset-based Hierarchical Clustering (FICH)" clustering algorithm to cluster Arabic. They conducted their experiments on 600 Arabic documents using N-grams based on word level, Trigrams and Quadgrams and they got promising results. They conducted their experiments using N-grams based on word level and character level Trigrams and Quadgrams. For the accuracy of clusters, word level outperforms both Quadgrams and Ttrigrams for both 4 and 6 natural classes, and Quadgrams gave better accuracy than Trigrams for both 4 and 6 natural classes. For the word level they got accuracy of 0.75 for four natural classes for 4 clusters, and they got accuracy of 0.70 for Quadgrams for four natural classes for 4 clusters, and 0.63 for Trigrams for four natural classes for 8 clusters.

2.3 Text Preprocessing in Document Clustering

Ahmed and Tiun[28] evaluated the efficiency and accuracy of Arabic Islamic document clustering based on K-means algorithm with three similarity/distance measures; Cosine, Jaccard similarity and Euclidean distance. Additionally, research investigated the effect of using stemming and without stemming words on the accuracy of Arabic Islamic text clustering. They used Islamic dataset (in-house). Based on the results, the K-means algorithm has the best results with Cosine similarity compared to Jaccard similarity and Euclidean distance. The results with Euclidean distance are better than the results with Jaccard similarity. In addition, they concluded that the results with stemming method are better than without stemming. They also depicted that the results depend on number of categories and size of dataset.

Froud et al. [29] proposed to compare the clustering results based on summarization with the full-text baseline on the Arabic Documents Clustering for five similarity/distance measures for three times: without stemming, and with stemming using Khoja's stemmer, and the Larkey's stemmer. They found that the Euclidean Distance, the Cosine Similarity and the Jaccard measures have comparable effectiveness for the partitional Arabic Documents Clustering task. They used the K-means algorithm as document clustering method. Results for Khoja's stemmer, the overall purity values for the

Euclidean Distance, the Cosine Similarity and the averaged Kullback-Leibler divergence (KL divergence) are quite similar and performs bad relatively to the other measures. Meanwhile, the Jaccard measure is the better in generating more coherent clusters with a considerable purity score. In this context, using the Larkey's stemmer, the purity value of the averaged KL Divergence measure is the best one with only 1% difference relatively to the other four measures. In the other hand, results without stemming shows the higher purity scores (0.77) than those shown for the Euclidean Distance, the Cosine Similarity and the Jaccard measures. In the other hand the Pearson Correlation and averaged KL Divergence are quite similar but still better than purity values for these measures KHOJA'S stemmer, and LARKEY'S stemmer. Other best results show the better and similar entropy values for the Euclidean Distance, the Cosine Similarity and the Jaccard measures. In overall results shows that the use of stemming affects negatively the clustering, this is mainly due to the ambiguity created when we applied the stemming (for example, two roots are obtained that made of the same letters but semantically different).

Froud et al. [30] evaluated the impact of the stemming on the Arabic Text Document Clustering. Their experiments show that the use of the stemming will not yield good results, but makes the representation of the document smaller and the clustering faster. The representation of the documents and the use of the stemming affect the final results. The stemming makes the representation of the document smaller and the clustering faster.

In **Osama and Wesam** [31], they evaluated stemming techniques in clustering of Arabic language documents and identified the most effective preprocessing approach for Arabic language, which is more complicated than most other languages. They used three stemming techniques: root-based stemming, light stemming, and without stemming. The data set used has been collected from BBC Arabic. The results indicate that the light stemming gets the best measurement values than without stemming and root-based stemming in Arabic document clustering. They applied feature selection methods and stemming techniques for Arabic text clustering. The data set was collected and classified manually into seven clusters: Middle East News, World News,

Business & Economy, Sports, International Press, Science & Technology, and Art & Culture. The testing dataset consists of 4,763 documents. Three stemming techniques have been used: without stemming which remains all terms, light stemming which removes common suffixes and prefixes, and root-based (Khoja) stemming which removes words have the same root. K-means was used to cluster the test documents; it was run for each technique of stemming individually. The experiments depicted that Light Stemming is the best technique for feature selection in Arabic language document clustering, but root based stemming get deterioration results for Arabic language document clustering; because Arabic language has a complex morphology, and it is a highly inflected language. The results of precision, recall and F-measure for three stemming cases: without stemming has values 0.6, 0.6 and 0.61 for precision, recall and F-measure respectively, the second type is light stemming and has values: 0.75, 0.7 and 0.72 for precision, recall and F-measure respectively, and the last type is root-based stemming and has values: 0.54, 0.53 and 0.54 for precision, recall and F-measure respectively. From results light stemming gets the best measurement values versus without stemming and root-based stemming in Arabic document clustering, because Arabic language has a complex morphology languages, and it is a highly inflected language, so root-based stemming gives backfire in clustering documents, but light stemming gives enhancement in clustering documents.

Al-Omari [1] evaluated and estimated the impact of stemming in clustering algorithm. The Arabic documents preprocessing which are used in his work are including; tokenization, stopword removal, and stemming function. The author used vector space model as the algorithm for clustering. The best result achieved was without stemming, and thus, it is evident that the results without stemming are better than with stemming. Their results give overall percent of successful documents without stemming equals to 0.69 while with stemming equals to 0.55. The experimental results showed that the clustering solution produced by the K-means algorithm is not stable; because of changing the initial k points every time the system is ran. In addition, the produced clusters facilitate examining each cluster for a clustering task. The task involves discriminating between successful and unsuccessful procedures. Furthermore,

experiments showed that K-means generally performed better if it selects several new centers during each iteration. Applying stemming on such clustering is not efficient because the documents must discriminate from each other to relate to the exact category; because the stemming is an abstract of word which leads to miss discriminating of documents.

Froud et al. [30] evaluated the impact of the stemming on the Arabic text document clustering. The dataset includes Corpus of Contemporary Arabic (CCA). The better results were achieved in their experiments without performing stemming on the dataset.

2.4 Similarity/Distance Measures in Document Clustering

Froud et al. [30] evaluated five similarity/distance measures: Euclidean Distance, Cosine Similarity, Jaccard Coefficient, Pearson Correlation Coefficient and Averaged Kullback-Leibler Divergence, for the testing dataset in the Arabic Text Document Clustering. They founded that the Euclidean Distance, the Cosine Similarity and the Jaccard measures have comparable effectiveness for the partitional Arabic Documents Clustering task. They have investigated that the Euclidean Distance, the Cosine Similarity and the Jaccard measures have comparable effectiveness for the partitional Arabic Documents.

CHAPTER 3: BACKGROUND OF DOCUMENT CLUSTERING

Clustering algorithms group a set of documents into subsets or clusters. The algorithms' goal is to create clusters that are coherent internally, but clearly different from each other. In other words, documents within a cluster should be as similar as possible; and documents in one cluster should be as dissimilar as possible from documents in other clusters. Clustering is the most common form of unsupervised learning. No supervision means that there is no human expert who has assigned documents to classes. In clustering, it is the distribution and makeup of the data that will determine cluster membership [32].

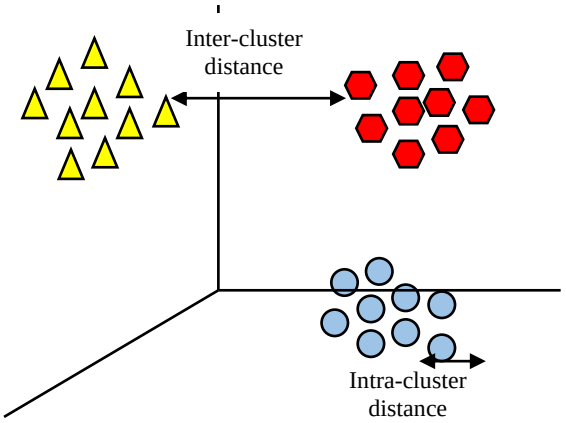

Figure 3.1: Three Clusters intra-cluster distances are minimized and inter-cluster distances are maximized

3.1 Document Clustering

Document clustering is an unsupervised learning task which aims at organizing documents into groups according to their similarity. Different aspects of similarity between documents can be defined. The most commonly-used aspect is the topic similarity, which is usually estimated based on the proximity of document vectors in the space of terms. Data clustering algorithms can be generally categorized into hierarchical and partitional[10]. Hierarchical clustering constructs a hierarchy of nested clusters, while partitional clustering divides data points into nonoverlapped clusters such that a specific criterion function is optimized[33]. The problem of document clustering is defined as follows. Given a set of n documents called DS, DS is clustered into a user-defined number of k document clusters DS_1, DS_2,…DS_k, (i.e. $\{DS_1, DS_2,\ldots DS_k\} = DS$) so that the documents in a document cluster are

similar to one another while documents from different clusters are dissimilar. In order to measure similarities between documents, documents have been represented based on the vector space model. In this model, each document *d* is represented as a high dimensional vector of words/terms frequencies (as the simplest form), where the dimensionality indicates the vocabulary of *DS*. Similarity between two documents has been traditionally measured by the cosine of the angle between their vector representations though there are a number of similarity measurements. Based on a cluster criterion function as an iterative optimization process that measures key aspects of intercluster and intra-cluster similarities, documents are grouped. A number of document clustering approaches have been developed for several decades. Most of these document clustering approaches are based on the vector space representation and apply various clustering algorithms to the representation[34]. The goal of a document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances (using an appropriate distance measure between documents). A distance measure (or, dually, similarity measure) thus lies at the heart of document clustering. The large variety of documents makes it almost impossible to create a general algorithm which can work best in case of all kinds of datasets[16].

The clustering of documents based on the similarity of their content may help to improve the search effectiveness[15]:

- **Improving Search Recall**

Standard search engines and IR systems return lists of documents that match a user query. It is often the case that the same concepts are expressed by different terms in different texts. For instance, a "car" may be called "automobile," and a query for "car" would miss the documents containing the synonym. However, the overall word contents of related texts would still be similar despite the existence of many synonyms. Clustering, which is based on this overall similarity, may help improve the recall of a query-based search in such a way that when a query matches a document its whole cluster can be returned. This method alone, however, might significantly degrade precision

because often there are many ways in which documents are similar, and the particular way to cluster them should depend on the particular query.

- **Improving Search Precision**

As the number of documents in a collection grows, it becomes a difficult task to browse through the lists of matched documents given the size of the lists. Because the lists are unstructured, except for a rather weak relevance ordering, he or she must know the exact search terms in order to find a document of interest. Otherwise, he or she may be left with tens of thousands of matched documents to scan. Clustering may help with this by grouping the documents into a much smaller number of groups of related documents, ordering them by relevance, and returning only the documents from the most relevant group or several most relevant groups. Experience, however, has shown that the user needs to guide the clustering process so that the clustering will be more relevant to the user's specific interest. An interactive browsing strategy called scatter/gather is the development of this idea.

- **Scatter/Gather**

The scatter/gather browsing method (Cutting et al. 1992; Hearst and Pedersen 1996) uses clustering as a basic organizing operation. The purpose of the method is to enhance the efficiency of human browsing of a document collection when a specific search query cannot be formulated. The method is similar to the techniques used for browsing a printed book. An index, which is similar to a very specific query, is used for locating specific information. However, when a general overview is needed or a general question is posed, a table of contents, which presents the logical structure of the text, is consulted. It gives a sense of what sorts of questions may be answered by more intensive exploration of the text, and it may lead to the particular sections of interest. During each iteration of a scatter/gather browsing session, a document collection is *scattered* into a set of clusters, and the short descriptions of the clusters are presented to the user. Based on the descriptions, the user selects one or more of the clusters that appear relevant. The selected clusters are then *gathered* into a new subcollection with which the process may be repeated. In

a sense, the method dynamically generates a table of contents for the collection and adapts and modifies it in response to the user's selection.

- **Query-Specific Clustering**

Direct approaches to making the clustering query-specific are also possible. The hierarchical clustering is especially appealing because it appears to capture the essense of the cluster hypothesis best. The most related documents will appear in the small tight clusters, which will be nested inside bigger clusters containing less similar documents. The work described in Tombros, Villa, and Rijsbergen (2002) tested the cluster hypothesis on several document collections and showed that it holds for query-specific clustering. Recent experiments with cluster-based retrieval (Liu and Croft 2003) using language models show that this method can perform consistently over document collections of realistic size, and a significant improvement in document retrieval can be obtained using clustering without the need for relevance information from by the user.

Document clustering is an effective approach to manage information overload. Documents can be clustered, i.e. grouped into sets of similar documents, with the help of human editors or automatically with the help of a computer program. Examples of manual clustering of websites, each a collection of documents, can be found in Yahoo![35] and Open Directory Project [36]. In these examples one can see that websites are grouped into broad topics and narrower subtopics within each broad topic, as opposed to many groups at the same level. Attempts at manual clustering of web documents are limited by the number of available human editors. For example, although the Open Directory Project has 67,026 editors to file a submitted website into the right category, the average wait time of a newly submitted site before it enters the appropriate category could be up to two weeks. A more efficient approach would be to use a machine learning algorithm to cluster similar documents into groups that are easier to grasp by a human observer. Two examples of such use of automated clustering are Vivisimo [37] and Google News[38]. Vivisimo offers an application that can be used to cluster results obtained from a search engine as a response to a query. This clustering

is done based on the textual similarity among result items and not based on the images or the multimedia components contained in them. Therefore, this type of clustering is known as text clustering or text document clustering. An example of Vivisimo clustering is shown in Figure 3.2. In this example the Vivisimo search engine was queried for ‘‘document clustering”. The returned results are grouped into clusters labeled ‘‘Methods”, ‘‘Information Retrieval”, and ‘‘Hierarchical, ‘‘Engine” etc. Thus a user interested in ‘‘hierarchical clustering” of documents can browse the results in the ‘‘Hierarchical” group. Note that in this example of document clustering there is no hierarchy of clusters, i.e., all the clusters are at the same level. On the other hand, Google News collects news articles from about 4500 sources and automatically clusters them into different groups such as ‘‘World”, ‘‘U.S.”, ‘‘Business”, ‘‘Sci/Tech”, ‘‘Sports”, ‘‘Entertainment”, and ‘‘Health” (Figure 3.3). Inside each group the articles are grouped together according to the event they describe[39].

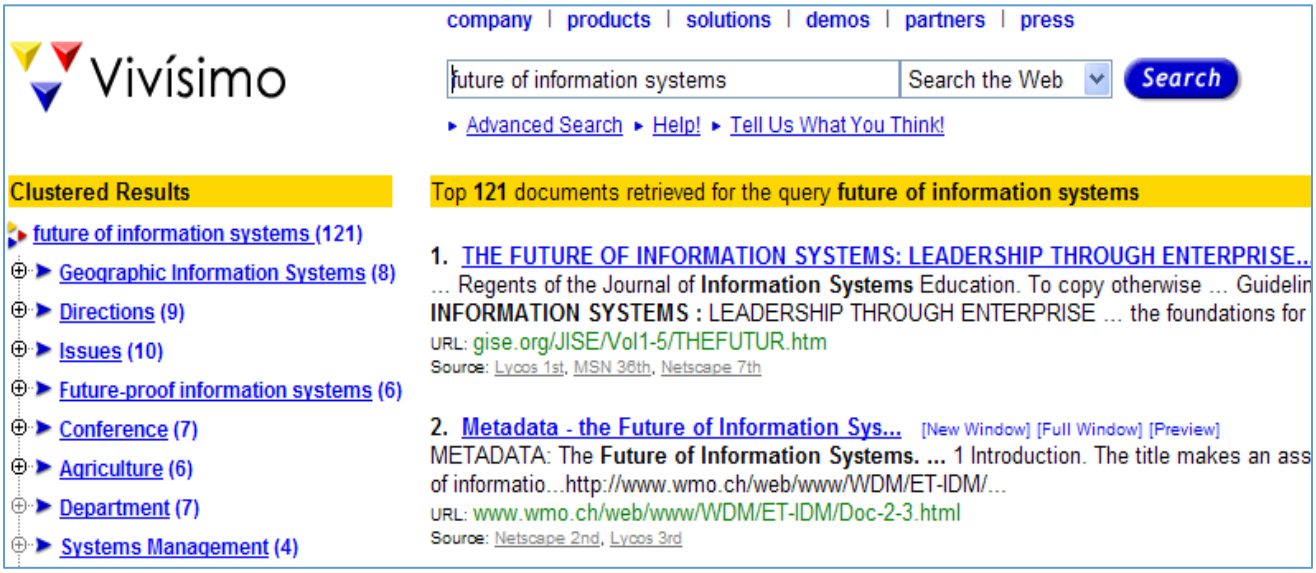

Figure 3.2: Vivisimo clustering solution [37]

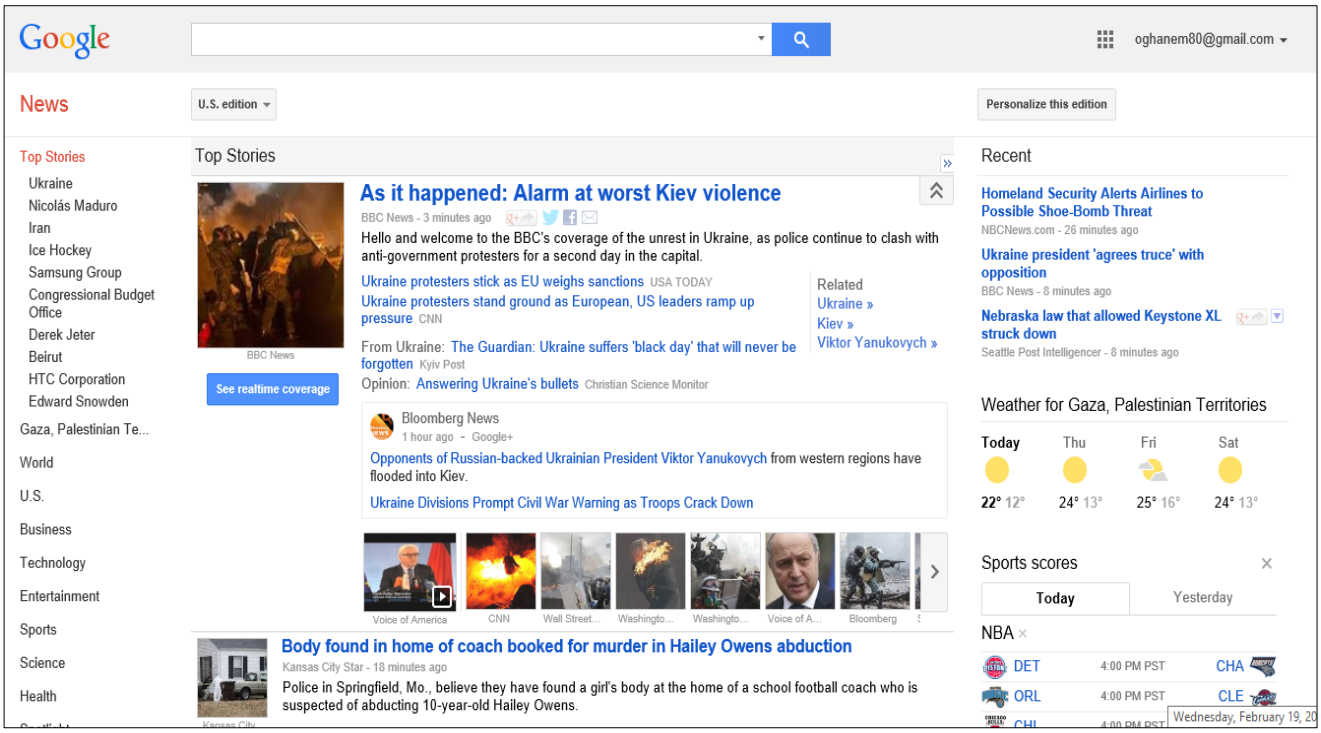

Figure 3.3: Google news clustering solution [38]

3.1.1 Document Clustering Applications

Document clustering is applied in many fields of business and science. Initially, document clustering was studied for improving the precision or recall in information retrieval systems. Document clustering has also been used to automatically generate hierarchical clusters of documents[40]. Following are few applications of document clustering [16]:

1. **Finding Similar Documents:** To find similar documents matching with the search result document. Clustering is able to discover documents that are conceptually alike compared to search-based approaches which discover documents sharing many of the same words.
2. **Organizing Large Document Collections:** To organize large number of uncategorized documents in taxonomy identical to the one human would create for easy retrieval.
3. **Duplicate Content Detection:** In many applications there is a need to find duplicates in a large number of documents. Clustering is employed for plagiarism detection, grouping of related news stories and to reorder search results rankings.
4. **Recommendation System**: Here, a user is recommended articles based on the articles the user has already read. Again this is possible by clustering of the articles, and improving the quality.
5. **Search Optimization**: Clustering helps a lot in improving the quality and efficiency of search engines as the user query can be first compared to the clusters instead of comparing it directly to the documents. Clustering is used in organizing the results returned by a search engine in response to a user's query [6]. Following this principle of cluster-based browsing by automatically organizing search results into meaningful categories are Teoma, vivisimo clustering engine, MetaCrawler, WebCrawler [41].

3.1.2 Document Clustering Procedure [42]

It is important to emphasize that getting from a collection of documents to a clustering of the collection, is not merely a single operation. It involves

multiple stages; which generally comprise three main phases: feature extraction and selection, document representation, and clustering.

Feature extraction begins with the parsing of each document to produce a set of features and exclude a list of pre-specified stop words which are irrelevant from semantic perspective. Then representative features are selected from the set of extracted features [13]. Feature selection is an essential preprocessing method to remove noisy features. It reduces the high dimensionality of the feature space and provides better data understanding, which in turn improves the clustering result, efficiency and performance. It is widely used in supervised learning, such as text classification[43]. Thus, it is important for improving clustering efficiency and effectiveness. Commonly employed feature selection metrics are term frequency (TF), inverse document frequency (TF- IDF), and their hybrids.

In the document representation phase, each document is represented by k features with the highest selection metric scores according to top-k selection methods. Document representation methods include binary (presence or absence of a feature in a document), TF (i.e., within-document term frequency), and TF-IDF. In the final phase of document clustering, the target documents are grouped into distinct clusters on the basis of the selected features and their respective values in each document by applying clustering algorithms[41].

3.1.2.1 Term Frequency–Inverse Document Frequency (TF-IDF)

In most clustering algorithms, the dataset to be clustered is represented as a set of vectors $X=\{x_1, x_2, \ldots, x_n\}$, where the vector x_i is called the feature vector of single object. In Vector Space Model (VSM), the content of a document is formalized as a dot in the multidimensional space and represented by a vector d, such as $d= \{w_1,w_2,.....,w_n\}$, where w_i is the term weight of the term t_i in one document. The term weight value represents the significance of this term in a document. To calculate the term weight, the occurrence frequency of the term within a document and in the entire set of documents is considered. The most widely used weighting scheme combines the Term

Frequency with Inverse Document Frequency (TF-IDF)[44]. The term frequency gives a measure of the importance of the term within the particular document. TF-IDF is a statistical measure which presents how important a word is to a document. More frequent words in a document are more important, i.e. more indicative of the topic[45].

Let f_{ij} = frequency of term *i* in document *j*

Now normalize term frequency (TF) across the entire corpus:

$$\mathrm{TF}_{ij} = f_{ij} / \max\{f_{ij}\} \quad (3.1)$$

The inverse document frequency is a measure of the general importance of the term. Terms that appear in many different documents are less indicative of overall topic.

Let df_i = document frequency of term i

= number of documents containing term i

IDF_i = inverse document frequency of term i,

$$\mathrm{IDF}_i = \log_2(\mathrm{N}/\, df_i) \quad (3.2)$$

Where N: total number of documents

A typical combined term importance indicator is TF-IDF weighting:

$$\mathrm{W}_{ij} = \mathrm{TF}_{ij} \times \mathrm{IDF}_i = \frac{\mathrm{F}_{ij}}{\max\{f_{ij}\}} \times \log_2(\mathrm{N}/\, df_i) \quad (3.3)$$

3.1.2.2 Dimension Reduction

Dimension reduction for large-scale text data is attracting much attention nowadays because high dimensionality causes serious problem for the efficiency of most of the algorithms [46]. These algorithms are of two types: feature extraction and feature selection. In the feature extraction, new features are combined from their original features through algebraic transformation. Though effective, these algorithms introduce high computational overhead,

making it difficult for real-world text data. In feature selection, subsets of features are selected directly. These algorithms are widely used in real-world tasks due to their efficiency, but are based on greedy strategies rather than optimal solutions [42].

3.1.3 Challenges in Document Clustering

Document clustering is being studied from many decades but still it is far from a trivial and solved problem. The challenges are:

1. Selecting appropriate features of the documents that should be used for clustering.
2. Selecting an appropriate similarity measure between documents.
3. Selecting an appropriate clustering method utilising the above similarity measure.
4. Implementing the clustering algorithm in an efficient way that makes it feasible in terms of required memory and CPU resources.
5. Finding ways of assessing the quality of the performed clustering.
6. Problem representation, including feature extraction, selection, or both.
7. Definition of proximity measure suitable to the domain.
8. Actual clustering of objects.
9. Data abstraction.
10. Evaluation.

Furthermore, with medium to large document collections (10,000+ documents), the number of term-document relations is fairly high (millions+), and the computational complexity of the algorithm applied is thus a central factor in whether it is feasible for real-life applications. If a dense matrix is constructed to represent term-document relations, this matrix could easily become too large to keep in memory - e.g. 100, 000 documents × 100, 000 terms = 1010 entries ~ 40 GB using 32-bit floating point values. If the vector model is applied, the dimensionality of the resulting vector space will likewise be quite high (10,000+). This means that simple operations, like finding the

Euclidean distance between two documents in the vector space, become time consuming tasks [15, 16].

3.1.4 Document Clustering Techniques

Several different variants of an abstract clustering problem exist. A *flat (or partitional)* clustering produces a single partition of a set of objects into disjoint groups, whereas a *hierarchical* clustering results in a nested series of partitions. Each of these can either be a *hard* clustering or a *soft* one. In a hard clustering, every object may belong to exactly one cluster. In soft clustering, the membership is fuzzy – objects may belong to several clusters with a fractional degree of membership in each. Irrespective of the problem variant, the clustering optimization problems are computationally very hard. The brute-force algorithm for a hard, flat clustering of n-element sets into k clusters would need to evaluate $k^n / k!$ possible partitionings. Even enumerating all possible single clusters of size l requires $n!/l!(n - l)!$, which is exponential in both n and l. Thus, there is no hope of solving the general optimization problem exactly, and usually some kind of a greedy approximation algorithm is used. *Agglomerative* algorithms begin with each object in a separate cluster and successively merge clusters until a stopping criterion is satisfied. *Divisive* algorithms begin with a single cluster containing all objects and perform splitting until a stopping criterion is met. "*Shuffling*" algorithms iteratively redistribute objects in clusters. The most commonly used algorithms are the K-means (hard, flat, shuffling), the EM-based mixture resolving (soft, flat, probabilistic), and the HAC (hierarchical, agglomerative) [15].

3.1.4 .1 Hierarchical Algorithms

Hierarchical techniques produce a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level). The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendogram. This tree graphically displays the

merging process and the intermediate clusters. For document clustering, the dendogram provides a taxonomy, or hierarchical index.

There are two basic approaches to generating a hierarchical clustering:

a) **Agglomerative:** Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.

b) **Divisive:** Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split[40].

3.1.4.2 Partitional Algorithms

Partitional clustering methods iteratively generate a single partition of the data, whereby the objective function is defined by the sum of distances from the pixel vector to the cluster prototype in n-dimensional space [47]. Hard partitional clustering is the case where each data point is assigned to one and only one cluster[47]. In fuzzy partitional clustering, each pixel is assigned a degree of membership of between 0 and 1 to each cluster [48]. Partitional algorithms used for document clustering include, but are not limited to: the k-means algorithm [10], spectral clustering [49], and non-negative matrix factorization [50]. The k-means algorithm [10] is the most widely used algorithm for data clustering. The goal of the algorithm is to group data points into k clusters such that the Euclidean distances between data points in each cluster and its centroid are minimized. Spherical k-means [51]is a variant of the basic k-means algorithm that uses cosine similarity between data points instead of the Euclidean distance. Spherical k-means is usually used with document data sets where the cosine similarity is a measure more indicative of proximity between documents.

3.1.4.3 Partitional Versus Hieratical Algorithms

Omaia M. Al-Omari [1] made a comparison between using partitional and hieratical algorithms in document clustering as follows:

The authors in (Yoo and Hu, 2006) performed a comprehensive comparison study of various document-clustering approaches such as K-means and Suffix Tree Clustering in terms of the efficiency, the effectiveness, and the scalability. They found that the partitional clustering algorithms are the most widely used algorithms in document clustering.

The work in (Kanungo and Mount, 2002), presented an implementation of a filtering K-means clustering algorithm. It established the practical efficiency of the filtering algorithm by presenting a data-sensitive analysis of the algorithm's running time. For the running time experiments, they used two algorithms, simple brute-force algorithm which computes the distance from every data point to every center. The second algorithm, called kd-center, operates by building a kd-tree with respect to the center points and then uses the kd-tree to compute the nearest neighbor for each data point. The results showed that the filtering K-means clustering algorithm runs faster as the separation between clusters increases.

The authors in (Zhong and Ghosh, 2002) focused on model-based partitional clustering algorithms because, according to the authors, many advantages provided. First, the complexity is O(n), where n is the number of data documents. In similarity-based approaches, calculating the pair wise similarities requires $O(n^2)$ time. Second, each cluster is described by a representative model, which provides a richer interpretation of the cluster.

As shown these researches indicates that partitional algorithms are more appropriate than hierarchal algorithms for document clustering. So in our experiments partitional algorithms will be used.

3.1.4.4 K-Means Clustering Algorithm

K-means algorithm is used in our experiments to get the best clustering results. It follows a simple and easy way to classify a given document set through a certain number of clusters (assume k clusters). The main idea is to define k centroids, one for each cluster. The simple K-means algorithm chooses the centroid randomly from the document set. The next step is to take each document belonging to a given data set and associate it to the nearest centroid. The K-means clustering partitions a data set by minimizing a sum of-squares cost function.

$$\boldsymbol{J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2} \qquad (3.4)$$

Where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a document $x_i^{(j)}$ and the cluster center c_j, is an indicator of the distance of the *n* documents from their respective cluster centroids.[52]

The *K*-Means Algorithm [52]

K-MEANS ($\{\overrightarrow{x_1}, \ldots., \overrightarrow{x_N}\}$,K)

1 $(\overrightarrow{s_1}, \overrightarrow{s_2} \ldots., \overrightarrow{s_K}) \leftarrow SELECTRANDOMSEEDS\ (\{\overrightarrow{x_1}, \ldots., \overrightarrow{x_N}\}, K)$

2 for k ←1 to K

3 do $\vec{\mu}_K \leftarrow \overrightarrow{s_K}$

4 while stopping criterion has not been met

5 do for k ←1 to K

6 do $w_k \leftarrow \{\}$

7 for n ←1 to N

8 do $j \leftarrow argmin_{j'} \left|\overrightarrow{\mu_{j'}} - \overrightarrow{x_n}\right|$

9 $w_j \leftarrow w_j \cup \{\overrightarrow{x_n}\}$ (reassignment of vectors)

10 for k ←1 to K

11 do $\vec{\mu}_K \leftarrow \frac{1}{|w_k|}\sum_{\vec{x}\in w_k}\vec{x}$ (recomputation of centroids)

12 return $\{\overrightarrow{\mu_1}, \ldots., \overrightarrow{\mu_K}\}$

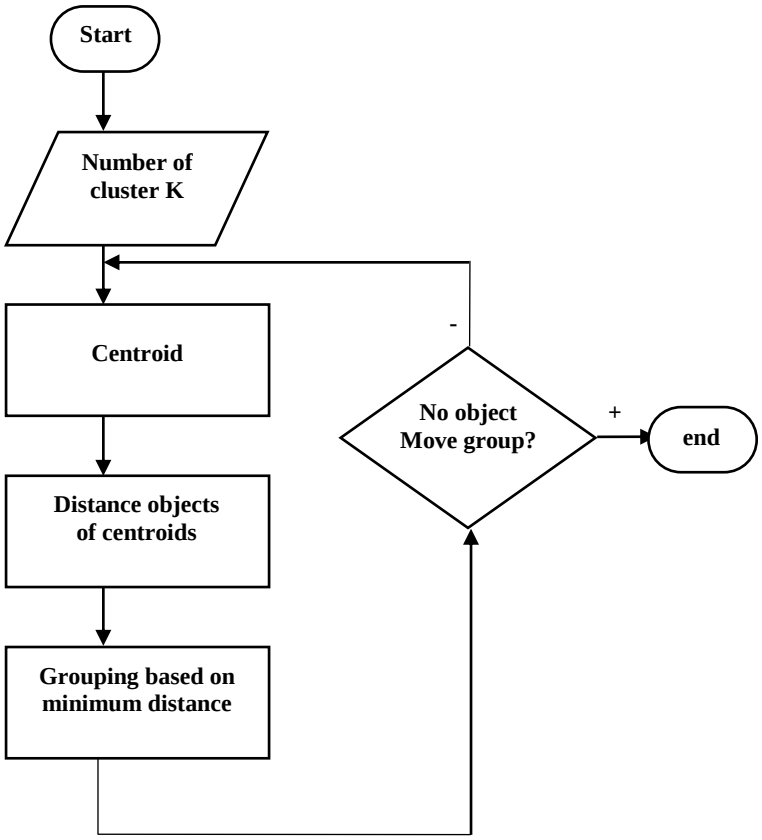

Figure 3. 4: The K-means algorithm Flow chart

As shown in Figure 3.4, K-means algorithm use cluster centroid to represent cluster, the first step is assigning data elements to the closest cluster. The second step is moving each centroid to its cluster. Repeat these steps until no change in movement of centroids.

3.1.4.5 Expectation Maximization (EM) Algorithm

The EM algorithm fall within a subcategory of the flat clustering algorithms, called Model-based clustering. The model-based clustering assumes that data were generated by a model and then tries to recover the

original model from the data. This model then defines clusters and the cluster membership of data. The EM algorithm is a generalization of K-Means algorithm in which the set of K centroids as the model that generate the data. It alternates between an *expectation step*, corresponding to reassignment, and a *maximization step*, corresponding to recomputation of the parameters of the model [16]. In addition to using distance as the similarity measure, the Expectation Maximization (EM) algorithm uses probabilities to measure the similarities. It is assumed that the points of a cluster follow a certain distribution [8]. By assuming the parameters of the distribution of each cluster, EM utilizes probability to judge which cluster a data point should be assigned to. Algorithm EM then adjusts the parameters of each cluster's distribution according to the data points in that cluster. Next, it reassigns these points according to these new distributions. These iterations continue until the clustering results converge. For example, if the distribution of Cluster C_i follows a given probability density function (abbreviated as pdf) $fc_i(v)$, then the probability for a point

with position v to belong to this cluster is:

$$\boldsymbol{P}(\boldsymbol{C_i}|\boldsymbol{v}) = \frac{\boldsymbol{P}(\boldsymbol{v}|\boldsymbol{C_i}) \times \boldsymbol{P}(\boldsymbol{C_i})}{\boldsymbol{P}(\boldsymbol{v})} = \frac{\boldsymbol{P}(\boldsymbol{C_i})}{\boldsymbol{P}(\boldsymbol{v})} \boldsymbol{f}_{\boldsymbol{C_i}}(\boldsymbol{v}) \qquad (3.5)$$

If a point at location v is more likely to belong to Cluster C_i than to Cluster C_j, i.e., $\boldsymbol{P}(\boldsymbol{C_i}|\boldsymbol{v}) > \boldsymbol{P}(\boldsymbol{C_j}|\boldsymbol{v})$, then this point will be assigned to Cluster C_i [53].

3.2 Similarity Measures

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems. In general, similarity/distance measures map the distance or similarity between the symbolic description of two objects into a single numeric value, which

depends on two factors- the properties of the two objects and the measure itself [54].

There are many measures can be used in clustering algorithms: Euclidean Distance, Manhattan, Cosine Similarity, Jaccard Coefficient, and Pearson Correlation Coefficient. We will study Euclidean Distance, Manhattan distance.

3.2.1 Euclidean Distance Function

Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text. It satisfies all the metric conditions and therefore is a true metric. It is also the default distance measure used with the K-means algorithm [54].

The distance between two documents is defined as:

$$D(i,j) = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \left(x_{i2} - x_{j2}\right)^2 + \cdots + \left(x_{in} - x_{jn}\right)^2}, \quad (3.6)$$

Where $i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ are two n-dimensional data objects.

Euclidean Distance is a main measuring similarity function in our clustering experiments, because of widely of using it in document clustering .

3.2.2 Manhattan Distance Function

Manhattan (or city block) distance function is the distance between two points is the sum of the absolute differences of their coordinates. The function is defined as:

$$D(i,j) = \left|x_{i1} - x_{j1}\right| + \left|x_{i2} - x_{j2}\right| + \cdots + \left|x_{in} - x_{jn}\right| \quad (3.7)$$

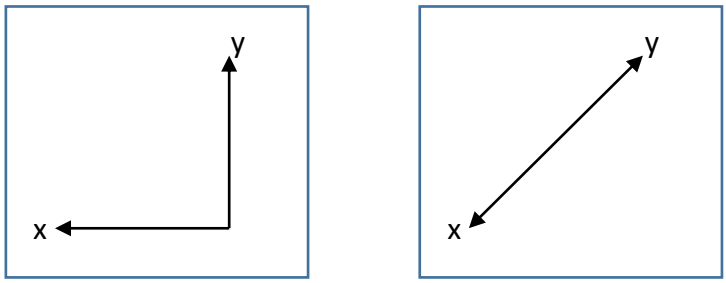

Figure 3. 5: Manhattan and Eculidean Distance

Figure 3.5 shows the geometric representations of Eculidean and Manhattan distance measures. As depicted Eculidean distance is defined as a straight line between two points, in the other hand Manhattan is defined as a distance between two points is the sum of the (absolute) differences of their coordinates.

CHAPTER 4: METHODOLOGY

In this chapter, we will discuss architecture of system for clustering Arabic text documents. Before clustering, text preprocessing will be applied to achieve the best results for clustering process, many preprocessing techniques have been used to enhance clustering results.

The architecture of clustering text is contains of:

1. Collect Arabic text documents
2. Apply text preprocessing for documents
3. Represent documents
4. Cluster documents
5. Evaluation results of clustering process

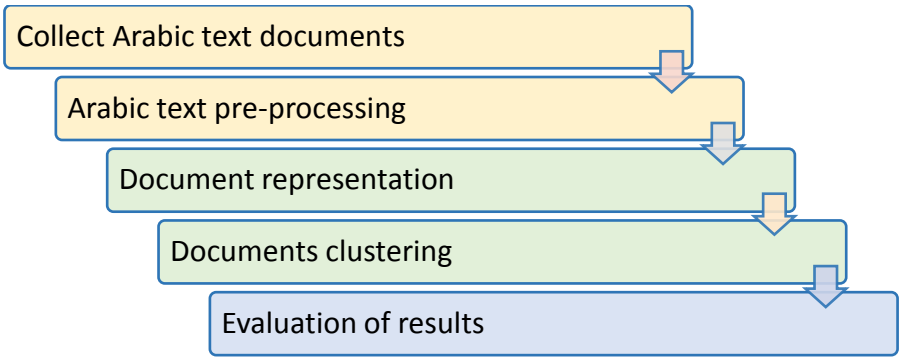

Figure 4. 1: Arabic document clustering architecture

4.1 Collect Arabic Text Documents

Collection of Data includes the processes like crawling, indexing, filtering, etc., These processes are used to collect documents to be clustered, indexed to store and retrieve in a better way, and filtered to remove extra data; for example, stopwords [42]. Large Arabic corpus of text documents as well as two freely public datasets were used for experiments. The first data set published by Saad in http://sourceforge.net/projects/ar-text-mining. The dataset was collected from CNN Arabic website because it is free, public, contains suitable number of documents for clustering process and suitable to for the hardware used in the experiments. CNN Arabic dataset used in the experiments is related to various categories, such as Business, Entertainments, Middle East News, Science and Technology, Sports, and World News.

Table 4.1 presents categories of CNN-Arabic corpus which includes 5070 documents, each document belongs to 1 of the 6 categories.

Table 4. 1: Number of documents in each category of CNN testing data set

id	Text Categories	Number of documents	% from corpus
1	Business	836	16.49%
2	Entertainments	474	9.35%
3	Middle East News	1462	28.84%
4	Science & Technology	526	10.37%
5	Sports	762	15.03%
6	World News	1010	19.92%
Total		**5,070**	**100%**

The second dataset used in the experiments was BBC Arabic corpus, which has been collected from BBC Arabic website bbcarabic.com. As shown in Table 4.2, the corpus includes 4,763 text documents, each text document belongs to 1 of 7 categories: Middle East News, World News, Business & Economy, Sports, International Press, Science & Technology and Art & Culture. The corpus contains 1,860,786 (1.8M) words and 106,733 district keywords after stopwords removal. The corpus was converted to utf-8 encoding and html tags were stripped.

Table 4.2: Number of documents in each category of BBC testing data set

id	Text Categories	Number of documents	% from corpus
1	Middle East News	2356	49.46 %
2	World News	1489	31.26 %
3	Business & Economy	296	6.21 %
4	Sports	219	4.59 %
5	International Press	49	1.028 %
6	Science & Technology	232	4.87 %
7	Art & Culture	122	2.56 %
Total		**4,763**	**100%**

4.2 Arabic Text Preprocessing Techniques

Text preprocessing consists of text input, word segment and stop-word filters, which require as much as 80 percent of the total effort. After the segment and filter, the dimensionality of the text feature vector can be significantly reduced, and hence, the processing effort needed in the discovery phase can be decreased greatly [55]. Preprocessing has been performed to represent the data in a form suitable for clustering. There are many ways of representing the documents, such as Vector-Model, graphical model, etc. Many measures were also used for weighing the documents and their similarities [42].

Arabic language consists of three types of words: nouns, verbs and particles. Nouns and verbs are derived from a limited set of about 10,000 roots (Darwish, 2002). Templates are applied to the roots in order to derive nouns and verbs by removing letters, adding letters, or including infixes. Furthermore, a stem may accept prefixes and/or suffixes in order to form the word (Darwish, 2003). The orientation of writing in Arabic is from right to left [56]. Viewing text as a Bag Of Tokens (BOT) (words, n-grams) is considered as one of widely used methods for text mining presentations, where

both classification and clustering can be applied. These are quite useful for mining and managing large volumes of text, however, there is a potential to do much more. The BOT approach loses a lot of information contained in text, such as word order, sentence structure and context. These are precisely the features that humans use to interpret text. Natural Language Processing (NLP) attempts to understand document completely (at the level of a human reader). General NLP is highly ambiguous. Natural Language is meant for human consumption and often contains ambiguities under the assumption that humans will be able to develop context and interpret the intended meaning [56-59]. Text processing includes tokenizing string to words, normalizing tokenized words, remove predefined set of words (stopwords), morphological analysis and finally term weighting [60, 61].

Preprocessing Techniques:

There are six techniques for Arabic text preprocessing:

1. String Tokenization
2. Dropping common terms: stop words
3. Normalization
4. Morphological Analysis Techniques (Stemming and Light Stemming)
5. Term Pruning
6. Vector Space Model (VSM) and Term Weighting Schemes

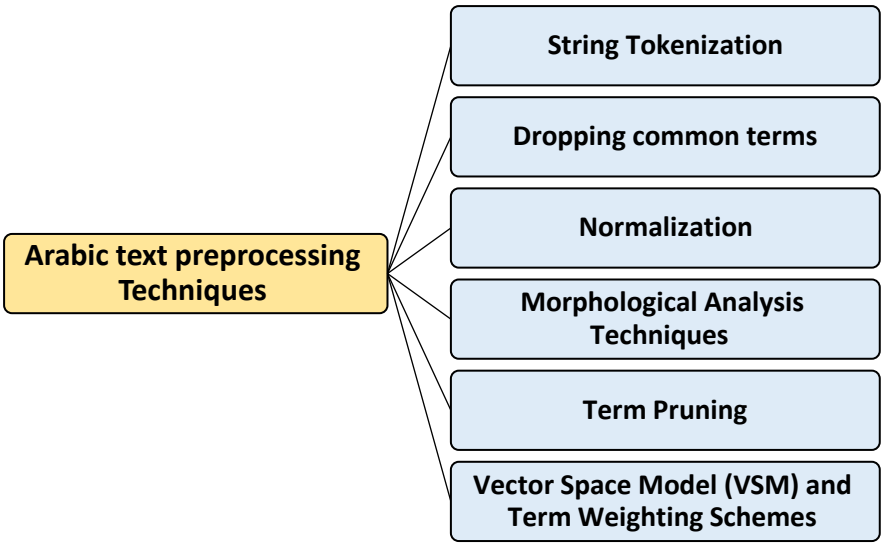

Figure 4. 2: Arabic text preprocessing Techniques

Figure 4.2 depicts the six techniques for Arabic text preprocessing to present the data for clustering process.

4.2.1 String Tokenization

One of the first steps of processing any text corpora is to divide the input text into proper units. These units could be characters, words, numbers, sentences or any other appropriate unit. The definition of a word here is not the exact syntactic form that is why we call it a 'token'. A token could refer to a syntactic word, a number or, as in Arabic, a whole grammatical phrase (e.g. (وسنساعدهم) "and we shall help them"). The process of extracting tokens is called tokenization (Attia, 2008; Lee et al, 2003) [62].Tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A type is the class of all tokens containing the same character sequence. A term is a (perhaps normalized) type that is included in the IR system's dictionary [32].

4.2.2 Dropping Common Terms: Stop Words

Stop words are common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. The general strategy for determining a stop list is to sort the terms by collection frequency (the total number of times each term appears in the document collection), and then to take the most frequent terms, often hand-filtered for their semantic content relative to the domain of the documents being indexed, as a stop list, the members of which are then discarded during indexing [32]. "stop-words," i.e., terms that are to be excluded from the indexing can be defined. Typically, a default list of English stop words includes "the", "a", "of", "since", etc., i.e., words that are used in the respective language very frequently, but communicate very little unique information about the contents of the document. For Arabic, stopwords list includes punctuations (? ! …), pronouns (... هما التي الذي هو), adverbs (...فوق تحت بينٌ), days of week (السبت الاحد الاثنينٌ ...), month of year (يناير فبراير مارس ...).

Stopwords list are removed because they do not help determining document topic and to reduce features [63].

4.2.3 Normalization

As data variables are of variable size and scales, it is therefore essential that we scale the data variables so that they are comparable. For example, if we have an age variable with a range from 0 to 100 and an income variable with a range from 30,000 to 100,000 thereby making it quite difficult to compare both variables. An increase of 10 corresponds to 10% in the age variable while accounting for only 0.01 % of the income variable. However if both variables are scaled to the same range of 0 and 1 then an increase in one variable would be directly comparable with the other variable. Data scaling can be performed by normalizing or standardizing the data variables, which is typically performed on the independent variables. Normalization scales each data variable into a range of 0 and 1 as shown in the following equation:

$$x_{ij}^{normalization} = \frac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}} \qquad (4.1)$$

where $x_{ij}^{normalization}$represents the normalized value, x_{ij} represents the value of interest, x_j^{min} represents the minimum value and x_j^{max} represents the maximum value. After being scaled, the minimum value would become 0 and the maximum value would become 1, while all other values would be in between 0 and 1 [64].

4.2.4 Morphological Analysis Techniques (Stemming and Light Stemming)

Stemming algorithms are needed in many applications such as natural language processing, compression of data, and information retrieval systems. In Arabic, the stemming approaches are applied in information retrieval field. Applying stemming algorithms as a feature selection method reduces the number of features since lexical forms (of words) are derived from basic building blocks; and hence, many features that are generated from the same stem are represented as one feature (their stem) [65]. Many stemmers have

been developed for English and other European languages. These stemmers mostly deal with the removal of suffixes as this is sufficient for most information retrieval purposes. Some of the most widely known stemmers for English are Lovins and Porter stemming algorithms [66]. The cause for needing special stemming algorithms for Arabic language can be described by El-Sadany and Hashish in the following points [67]:

i. Arabic is one of Semitic languages which differs in structure of affixes from Indo-European type of languages such as English and French;
ii. Arabic is mainly roots and templates dependent in the formation of words;
iii. Arabic roots consonants might be changed or deleted during the morphological process.

There are three different approaches for stemming: the root-based stemmer, the light stemmer, and the statistical stemmer. These stemming types are shown below in Figure 4.3.

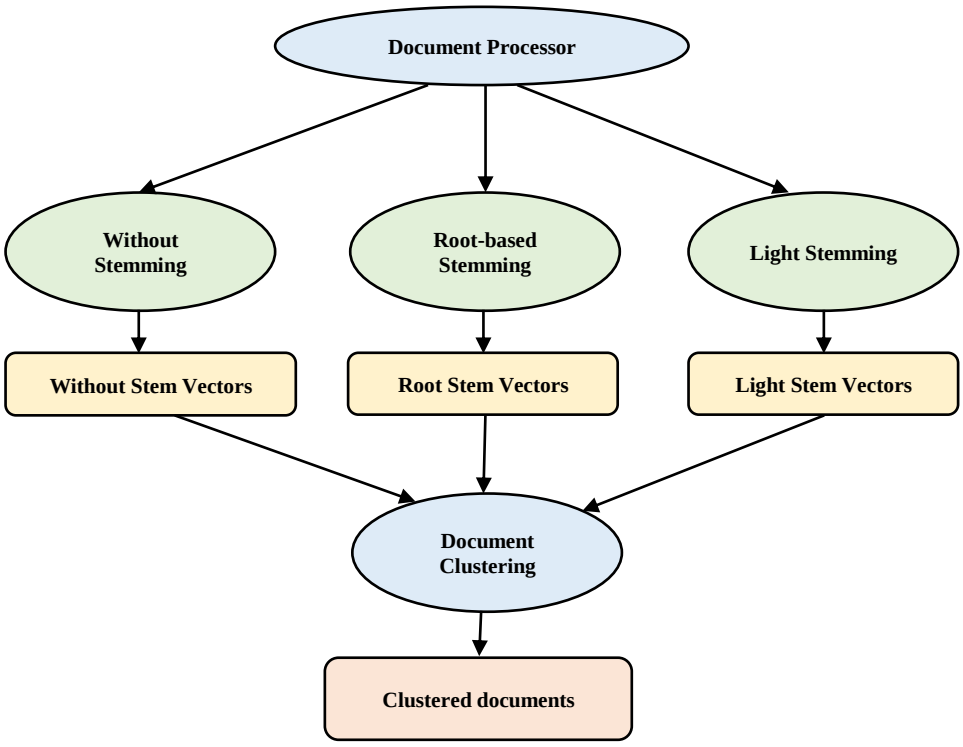

Figure 4. 3: Stemming System Architecture

Two types of stemming (root-based and light) will be applied to Arabic documents in addition to without stemming type:

a. Root-based Stemming

Stemming using root extractor which uses morphological analysis for Arabic words, Figure 4.4 depicts an example of using stemming for feature selection. Note that several words such as (الكتاب الكاتب المكتبة) which mean "the library", "the writer" and "the book" respectively are reduced to one stem (كتب) which means write [68] as shown in figure 4.4 [69], which describes preprocessing steps in root based stemming. Several algorithms have been developed for this approach such as:

RDI MORPHO3 Algorithm, Sebawai root extractor (SR) Algorithm, and Khoja Stemming Algorithm which will be used in our experiments.

- **RDI MORPHO3**

This system uses rules in conjunction with statistics in order to build a list of possible prefix-suffix template combinations (Attia, 2000). These combinations are used in order to transform the word to a root. The main disadvantage of this system is that the rules are built manually which is time consuming and demanding a deep knowledge of the Arabic language. The output of MORPHO3 system is a morphological analysis of the words including its root, stem, meaning of prefixes and suffix, etc…

- **Sebawai Root Extractor (SR)**

Sebawai is very similar to MORPHO3 root extractor. However, it uses automatic rules rather than manual rules (Darwish, 2003). Rules have been obtained through training the system with a list of word-root pairs. The author suggests obtaining the training list by three ways; (a) manual construction, (b) using another morphological analyzer tool such as MORPHO3, or (c) parsing a dictionary[70].

- **Khoja Stemming Algorithm**

Khoja and Garside developed stemmer algorithms [71]. The algorithm, developed by using both Java and C++ languages, removes the longest suffix and the longest prefix. It then matches the remaining word with verbal and noun patterns, to extract the root. The stemmer makes use of several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words. The algorithm achieves accuracy rates of up to 96%. The algorithm correctly stems most Arabic words that are derived from roots.

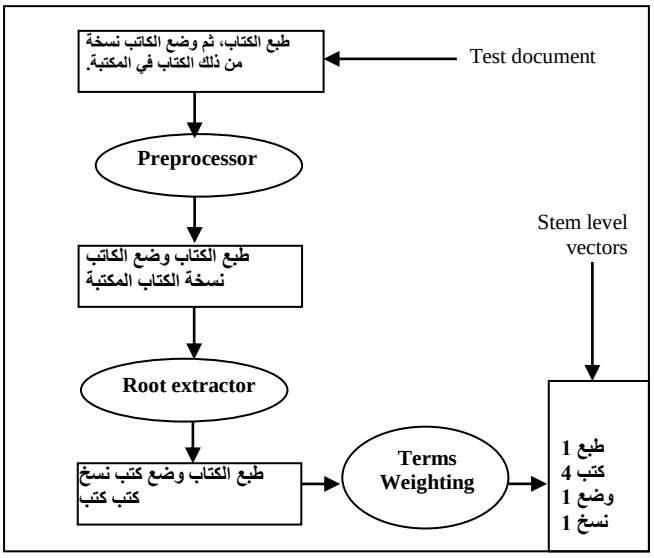

Figure 4.4: Preprocessing with root-based stemming

Algorithm 4.1: Arabic Stemming Algorithm Steps [71]

1. *Remove diacritics.*
2. *Remove stopwords, punctuation, and numbers.*
3. *Remove definite article (ال).*
4. *Remove inseparable conjunction (و).*
5. *Remove suffixes.*
6. *Remove prefixes.*
7. *Match result against a list of patterns.*
   - *If a match is found, extract the characters in the pattern representing the root.*
   - *Match the extracted root against a list known "valid" roots.*

8. *Replace weak letters (ي)(أ)(و) with (و).*
9. *Replace all occurrences of Hamza(ؤ)(ء)(ئ) with(ا).*
10. *Two letter roots are checked to see if they should contain a double character. If so, the character is added to the root.*

b. Light Stemming

The main idea for using light stemming is that many word variants do not have similar meanings or semantics. However; these word variants are generated from the same root. Thus, root extraction algorithms affect the meanings of words. Light stemming by comparison aims to enhance the categorization performance while retaining the words' meanings. It removes some defined prefixes and suffixes from the word instead of extracting the original root[72]. Light-stemming keeps the word's meanings unaffected. Figure 4.5 demonstrates an example of using light stemming. Here we note that light stemming maintains the difference between (الكاتبون الكتاب) which means "the book" and "the writers" respectively; their light stems are (كاتب كتاب) which means book and writer [73] .

Algorithm 4.2: Arabic Light Stemming Algorithm Steps [74]

1. *Normalize word:*
   - *Remove diacritics.*
   - *Replace(إ)(أ)(آ) with(ا).*
   - *Replace(ة) with(ه).*
   - *Replace(ى) with(ي).*

Stem prefixes:

2. *Remove prefixes: (و)، (ال)، (وال)، (بال)، (كال)، (فال)، (لل).*

Stem suffixes:

3. *Remove suffixes:(ي)، (ه)، (ية)، (ين)، (ون)، (ات)، (ان)، (ها) .*

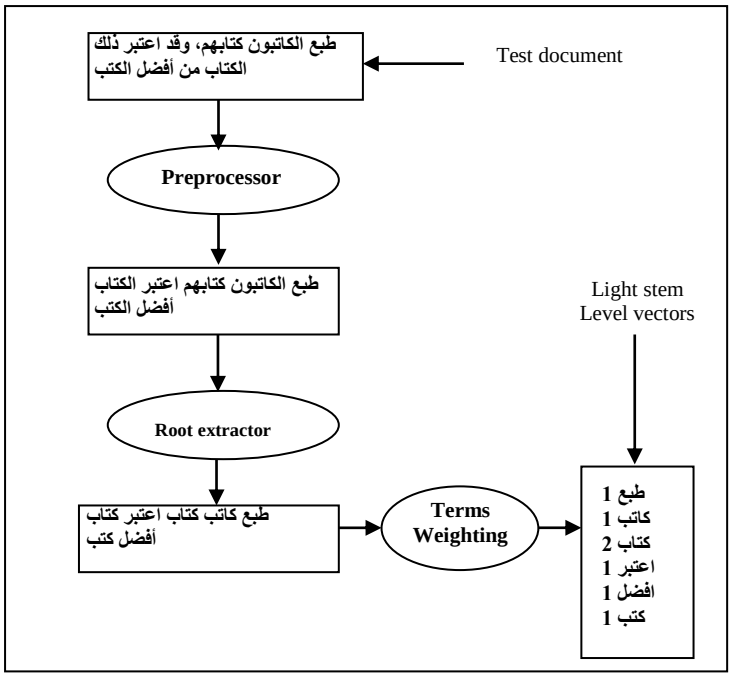

Figure 4.5: Preprocessing with light stemming

4.2.5 Term Pruning

Term Pruning, in Machine Learning, refers to an action of removing nonrelevant features from the feature space. In text mining, pruning is a useful preprocessing concept because most words in the text corpus are low-frequency words. According to the Zipf's law, given some corpus of natural language texts, if words are ranked according to their frequencies, the distribution of word frequencies is an inverse power law with the exponent of roughly one [75]. This implies that, in any training corpus, the majorities of the words in the corpus appear only a few times. A word that appears only a few times is usually statistically insignificant - low document frequency, low information gain, etc. Moreover, the probability of seeing word, that occurs only once or twice in the training data, in the future document is very low [76]. In the other hand term pruning can be defined as the process of eliminating the words that its count is less or greater than a specific threshold [63].

4.2.6 Vector Space Model (VSM) and Term Weighting Schemes

The representation of a set of documents as vectors in a common vector space is known as the vector space model and is fundamental to a host of information retrieval operations ranging from scoring documents on a query, document classification and document clustering. A pivotal step in this development is the view of queries as vectors in the same vector space as the document collection [32]. In the Vector Space Model, the contents of a document are represented by a multidimensional space vector. The proper classes of the given vector are determined by comparing the distances between vectors. The procedure of the Vector Space Model can be divided into three stages:

1. The first step is document indexing, when most relevant terms are extracted.
2. The second stage is based on the introduction of weights associated to index terms in order to improve the retrieval relevant to the user.
3. The last stage classifies the document with a certain measure of similarity.

The most common vector space model assumes that the objects are vectors in the high-dimensional feature space. A common example is the bag-of-words model of text documents. In a vector space model, the similarity function is usually based on the distance between the vectors in some metric.

In VSM, document can be represented as vector space in high dimensions. Each document can be represent as vector space *V(d)*, $V(d)=((t_1,w_1),(t_2,w_2),\ldots,(t_n,w_n))$.Where, t_i is the feature *i* in document d, w_i is the weight of t_i in document d. The value of w_i can be *0* or *1*, in the other hand *tf*idf* is a widely used method in term weight (w_i) calculation in document representation. For *tf*, reflects local weight in each document, *idf* reflects global weight in all documents [23].

Term weighting is one of preprocessing methods; used for enhanced text document presentation as feature vector. Term weighting helps us to locate important terms in a document collection for ranking purposes [77]. The

popular schemes for term weight are Boolean model, Term Frequency (TF), Inverse Document Frequency (IDF), and Term Frequency-Inverse Document Frequency (TF-IDF).

- **Boolean Model**

The Boolean model is the simplest retrieval model based on Boolean algebra and set theory [78]. Boolean model indicates to absence or presence of a word with Booleans 0 or 1 respectively [79].

- **Term Frequency**

This approach is to assign to each term in a document a weight for that term that depends on the number of occurrences of the term in the document. To get this compute a score between a query term t and a document d, based on the weight of t in d. The simplest approach is to assign the weight to be equal to the number of occurrences of term t in document d. This weighting scheme is referred to as term frequency Term Frequency and is denoted $TF_{t,d}$, with the subscripts denoting the term and the document in order [32].

$$TF(d, t_i) = \frac{n(d,t_i)}{\sum_i n(d,t_i)} \qquad (4.2)$$

Where $n(d,t_i)$ is the number of occurrences of t_i in a document and $\Sigma_i\, n(d, t_i)$ is the total number of tokens in document.

- **Inverse Document Frequency**

Raw term frequency suffers from a critical problem: all terms are considered equally important when it comes to assessing relevancy on a query. In fact certain terms have little or no discriminating power in determining relevance. For instance, a collection of documents on the auto industry is likely to have the term auto in almost every document. To this Inverse document frequency IDF(t) is scale down the terms that occur in many documents. We introduce a mechanism for attenuating the effect of terms that occur too often in the collection to be meaningful for relevance determination. An immediate idea is to scale down the term weights of terms with high

collection frequency, defined to be the total number of occurrences of a term in the collection. The idea would be to reduce the tf weight of a term by a factor that grows with its collection frequency. Instead, it is more commonplace to use for this purpose the document frequency dft, defined to be the number of documents in the collection that contain a term t. This is because in trying to discriminate between documents for the purpose of scoring it is better to use a document-level statistic (such as the number of documents containing a term) than to use a collection-wide statistic for the term [32].

$$IDF(t_i) = \log(\frac{D}{D_i}) \quad (4.3)$$

Where *Di* is the number of documents containing t_i and *D* is the total number of documents in the collection.

- **Term Frequency-Inverse Document Frequency (TF-IDF)**

Term Frequency and Inverse Document Frequency (TF-IDF), is a popular method of preprocessing documents in the information retrieval community [80].

$TF\text{-}IDF_{t,d}$ assigns to term *t* a weight in document *d* that is

1. highest when t occurs many times within a small number of documents (thus lending high discriminating power to those documents).
2. lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal).
3. lowest when the term occurs in virtually all documents [32].

$$w_{ij} = tfidf(t_i, d_j) = \frac{f_{ij}}{\sqrt{\sum_{k=1}^{M} {f_{kj}}^2}} \times \log\left(\frac{N}{n_i}\right) \quad (4.4)$$

Where *N* is the number of documents in the data set, M is the number of terms used in the feature space, f_{ij} is the frequency of a term *i* in document *j*, and n_i denotes the number of documents that term *i* occurs in at least once.

In thesis we will apply Term Frequency-Inverse Document Frequency (TF-IDF) preprocessing method to enhance text document presentation as feature vector.

4.3 Document Representation

The documents are represented by feature vectors. A feature is simply an entity without internal structure – a dimension in the feature space. A document is represented as a vector in this space – a sequence of features and their weights. The most common bag-of-words model simply uses all words in a document as the features, and thus the dimension of the feature space is equal to the number of different words in all of the documents. The methods of giving weights to the features may vary. The simplest is the binary in which the feature weight is either one – if the corresponding word is present in the document – or zero otherwise. More complex weighting schemes are possible that take into account the frequencies of the word in the document, in the category, and in the whole collection. The most common TF-IDF scheme gives the word w in the document d the weight [81]. This scheme is mentioned previously in details.

4.4 Documents Clustering

As mentioned in details previously in chapter 3. Clustering algorithms group a set of documents into subsets or clusters. The algorithms' goal is to create clusters that are coherent internally, but clearly different from each other. In other words, documents within a cluster should be as similar as possible; and documents in one cluster should be as dissimilar as possible from documents in other clusters. Clustering is the most common form of unsupervised learning. No supervision means that there is no human expert who has assigned documents to classes. In clustering, it is the distribution and makeup of the data that will determine cluster membership. The difference between clustering and classification may not seem great at first. After all, in both cases we have a partition of a set of documents into groups. But the two problems are fundamentally different. Classification is a form of supervised learning: its goal is to replicate a categorical distinction that a human

supervisor imposes on the data. In unsupervised learning, of which clustering is the most important example, there is no such teacher to guide. The key input to a clustering algorithm is the distance measure [32].

4.5 Document Clustering Tool (WEKA)

WEKA (Waikato Environment for Knowledge Analysis) is a data mining open-source tool in abroad, but it is rarely used at home. We provide documents preprocessing, and apply K-means algorithm in the Arabic document clustering by adjusting the parameters in WEKA. WEKA is a famous with data mining software and is well received in abroad [82]. For example, lots of document clustering and document categorization experiments have been carried out using 20 Newsgroups and Reuters-21578 corps based on WEKA [83]. The main functions of document clustering in WEKA include three aspects as below:

(1) Convert directory structure to arff file.
(2) Convert string attributes into a set of attributes representing word occurrence.
(3) Apply clustering algorithm.

WEKA is an open-source software, researchers can modify or add new algorithm when they needed [84]. Clustering tools and options using WEKA is depicted in Figure 4.6. Document representation using WEKA is depicted in Figure 4.7. String to Word Vector tools using WEKA, is depicted in Figure 4.8. Figure 4.10, depicts clustering options using WEKA, which can change clustering algorithm and change properties of algorithm.

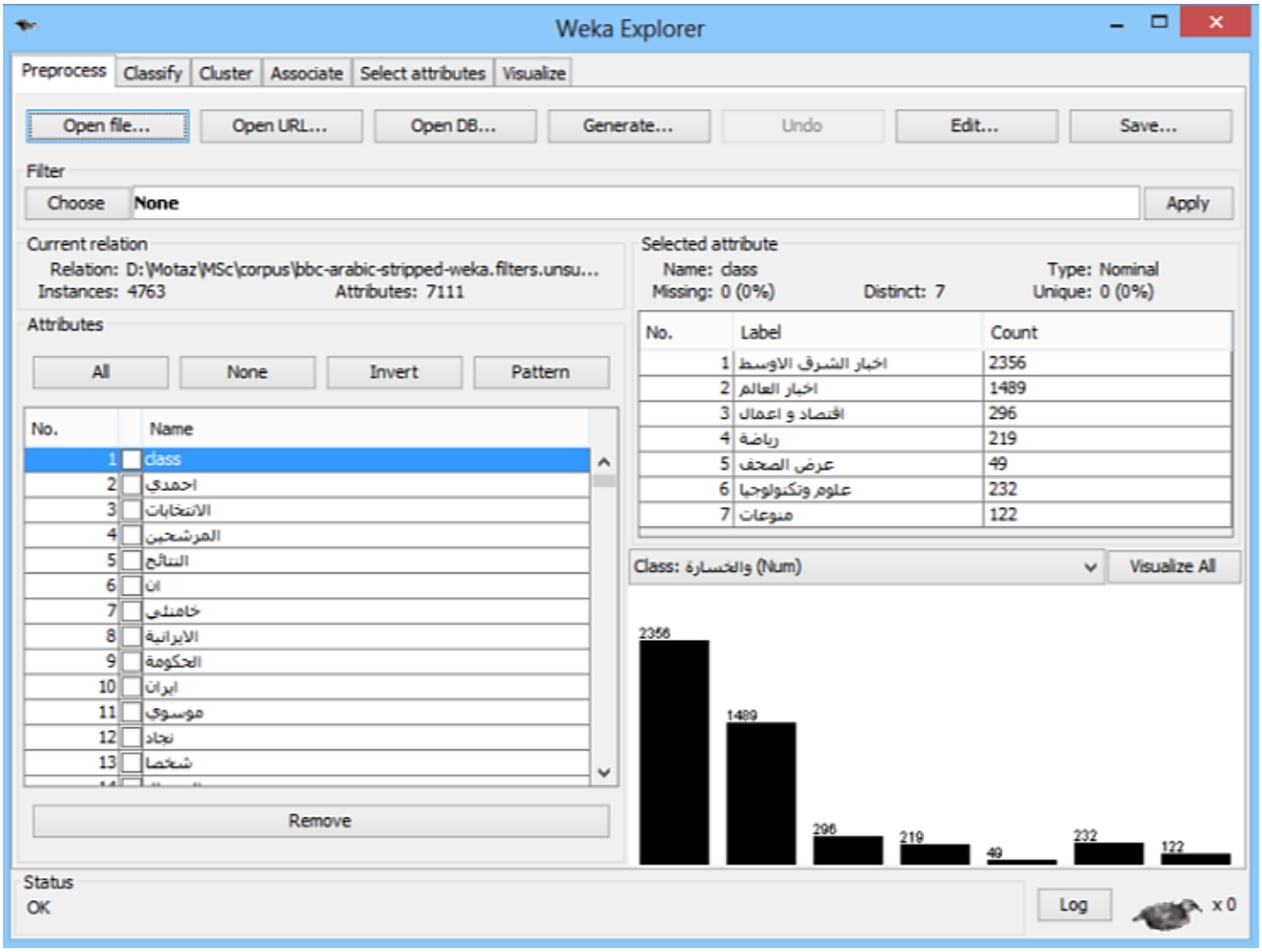

Figure 4.6: Clustering tools and options (WEKA)

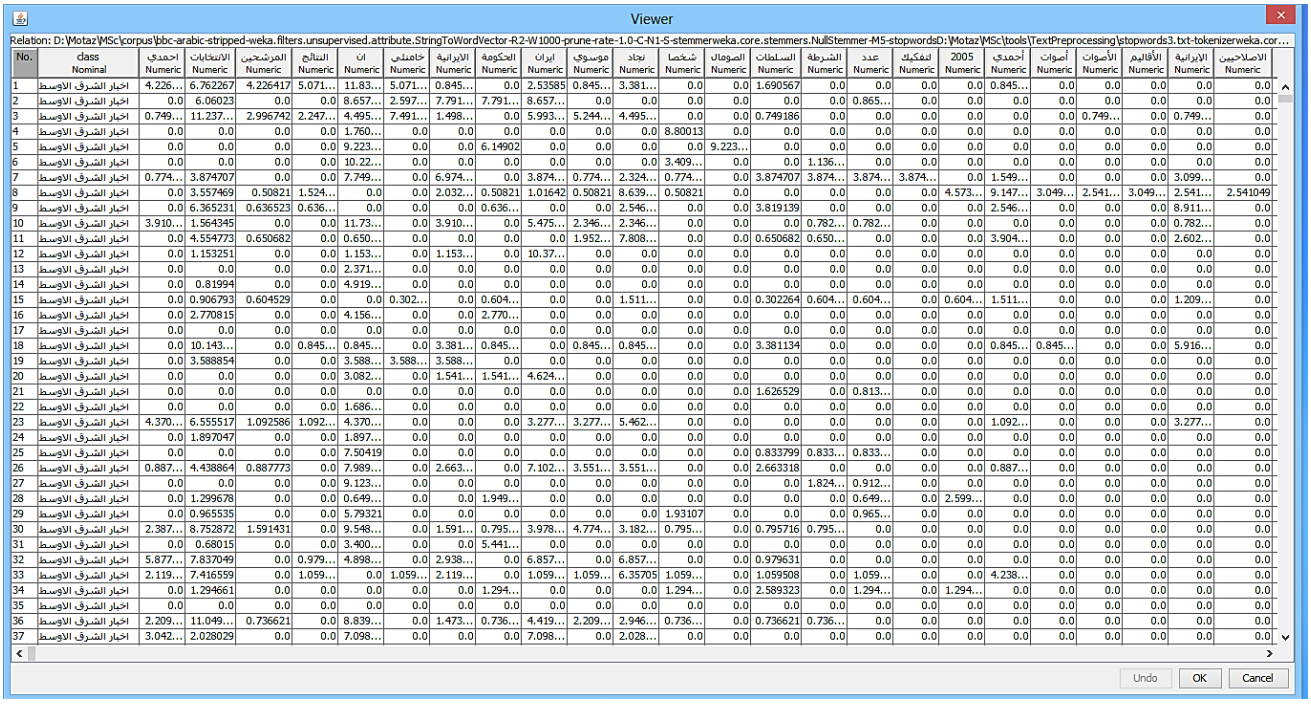

Figure 4.7: Document representation in WEKA

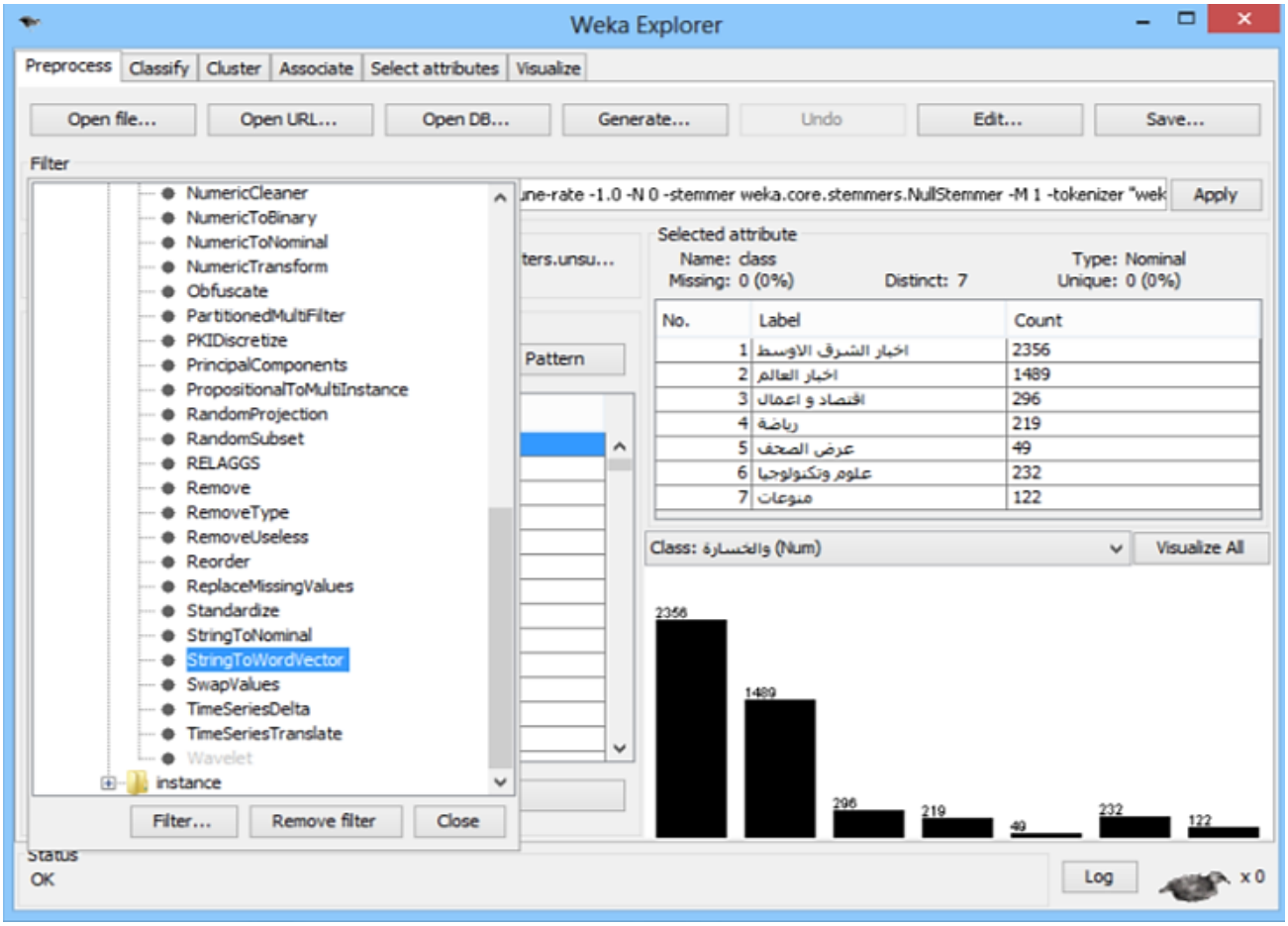

Figure 4. 8: String To Word Vector tools using WEKA

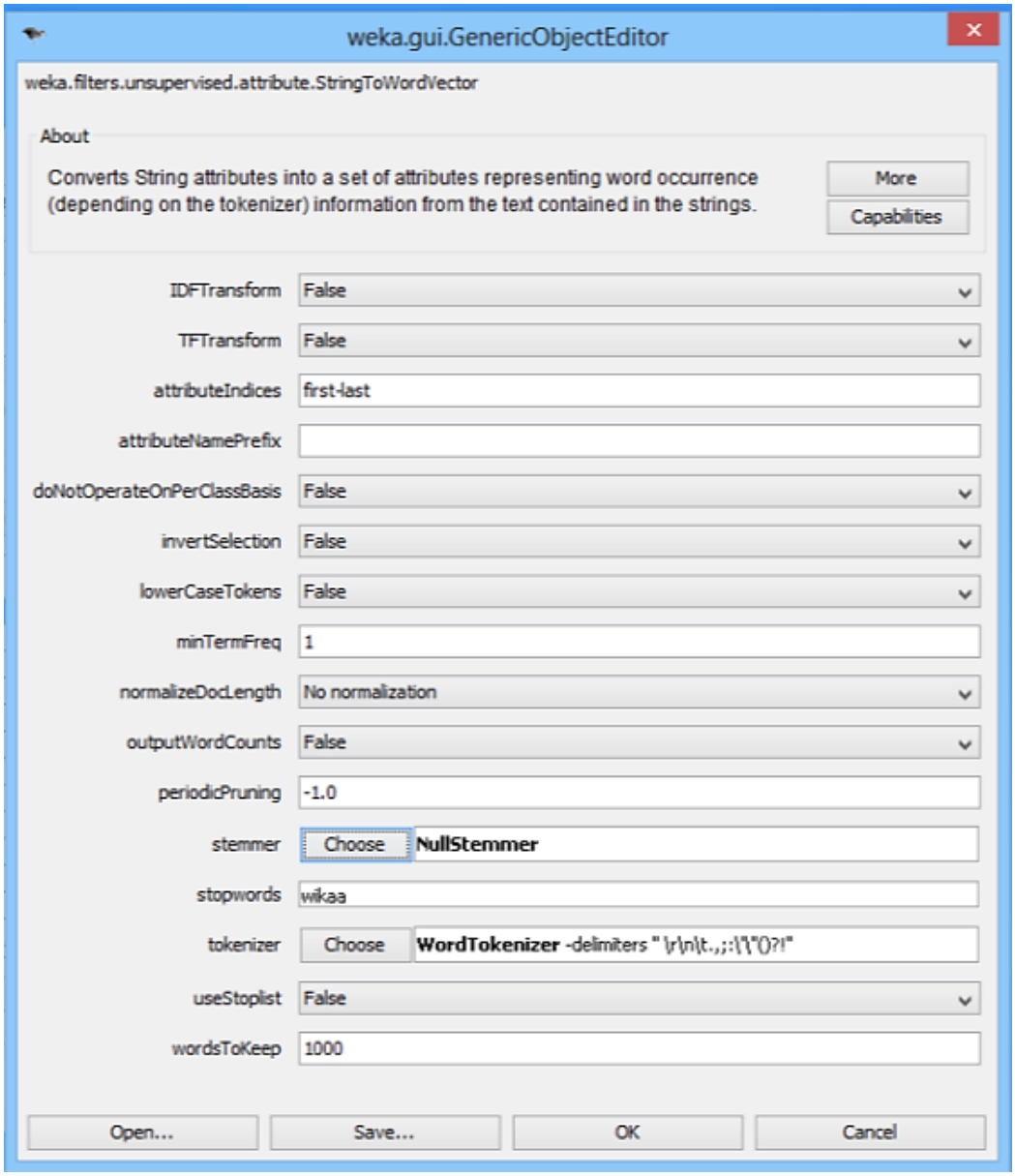

Figure 4. 9: Preprocessing options in WEKA

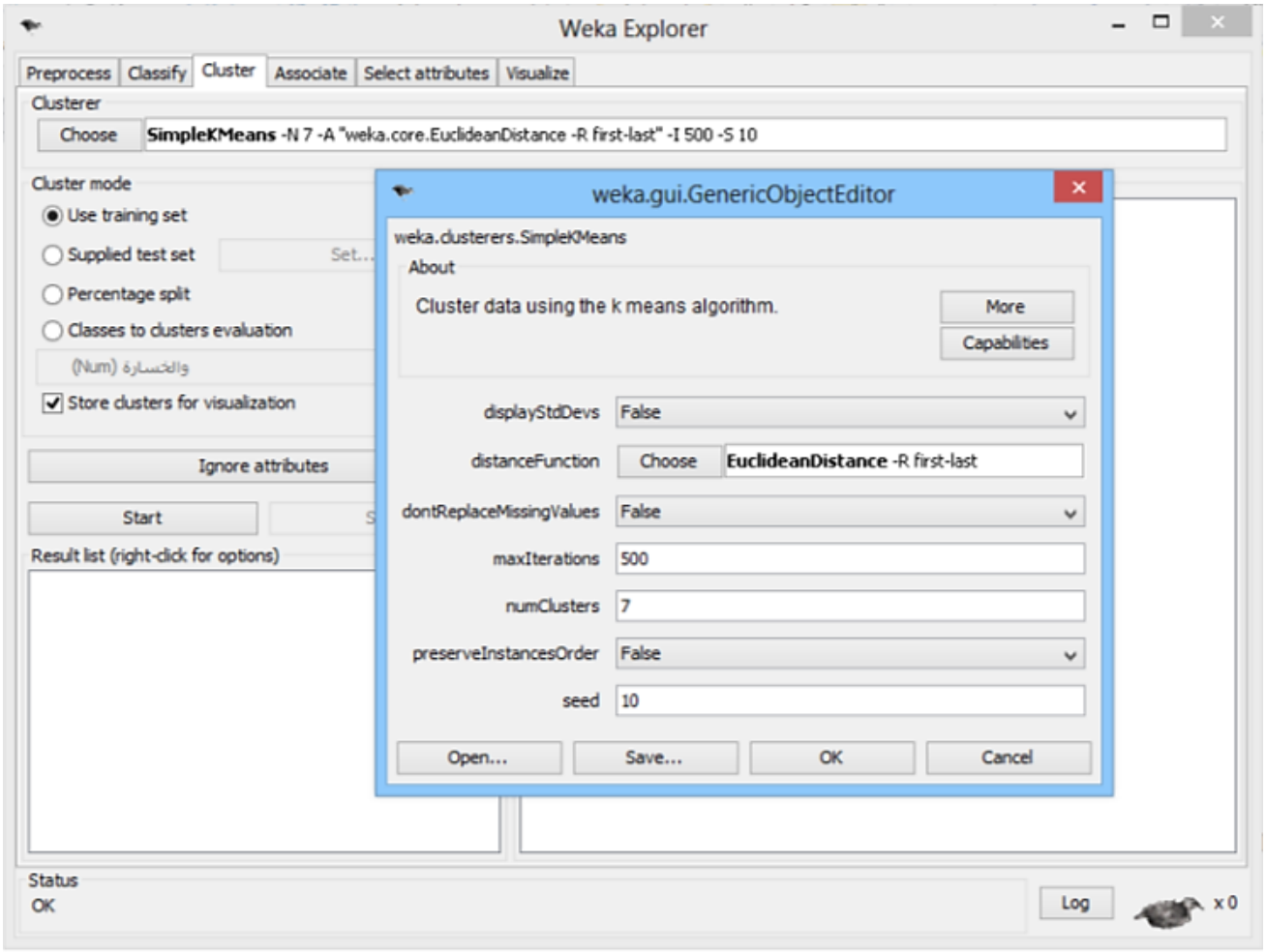

Figure 4. 10: Clustering options using WEKA

4.6 Evaluation

There are many evaluation standards in information retrieval used in document clustering such as Entropy, Cluster Purity, and F-measure which will be used in this thesis.

F-measure: F-measure [85] is widely used in text clustering. It provides a good balance between precision and recall, which is excellent in the context of information retrieval [86].

- Precision shows how many documents are in right cluster with respect to the cluster size.
- Recall shows how many documents are in the right cluster with respect to total documents.

Precision (P) is the fraction of retrieved documents that are relevant [32].

$$Precision = \frac{\#\,(relevant\ items\ retrieved)}{\#(retrieved\ items)} = \text{P(relevant|retrieved)} \qquad (4.5)$$

Recall (R) is the fraction of relevant documents that are retrieved.

$$Recall = \frac{\#\,(relevant\ items\ retrieved)}{\#(relevant\ \ items)} = \text{P}(retrieved\ |\text{relevant}) \qquad (4.6)$$

These notions can be made clear by examining the following contingency table:

	Relevant	**Non relevant**
Retrieved	True positive (tp)	False positive (fp)
Not retrieved	False negative (fn)	True negative (tn)

Then:

- Precision (P) = tp/(tp+fp) (4.7)
- Recall (R) = tp/(tp+fn) (4.8)

on the other hand we can compute precision and recall for class i and cluster j is defined as:

$$Recall\ (i,j) = \frac{n_{ij}}{n_j} \qquad (4.9\ )$$

$$Precision\ (i,j) = \frac{n_{ij}}{n_i} \qquad (4.10)$$

Where n_{ij} is the number of documents with class label i in cluster j , n_i is the number of documents with class label i, and n_j is the number of documents in cluster j , and n is the total number of documents.

The F-measure for class i and cluster j is given as:

$$F(i,j) = \frac{2 * Recall(i,j) * Precision\ (i,j)}{Recall(i,j) + Precision\ (i,j)} \tag{4.11}$$

Then total F-measure of clustering process is calculated as:

$$F = \sum {}^{n_i}/_{n * maxF(i,j)} \tag{4.12}$$

CHAPTER 5: EXPERIMENTAL RESULTS AND ANALYSIS

In this chapter we will discuss the experimental results of applying clustering technique in Arabic text documents with many text preprocessing methods and combinations. As mentioned in previous chapter, the data sets are: CNN Arabic corpus which includes 5,070 text documents, each text document belongs 1 of the 6 categories: Business, Entertainments, Middle East News, Science & Technology, Sports, and World News. The other dataset: BBC Arabic corpus which includes 4,763 documents, each document belongs to 1 of the 7 domains or categories: Middle East News, World News, Business & Economy, Sports, International Press, Science & Technology, and Art & Culture. WEKA data mining tool is used for text preprocessing and document clustering. Experiment environment as follows, operating system: Windows 8, CPU: Intel Core i7 2670QM 2.20 GHz, Memory: 8 GB, WEKA version: 3.6.4.

Experimental results were investigated by measuring evaluation of clustered documents in many cases of preprocessing techniques. The two most frequent and basic measures for information retrieval effectiveness (measuring precision and recall) [32] were used for accuracy reasons. The other measurement is F-Measure which is a single measure that trades off precision versus recall [32]. The impact of the following text preprocessing techniques will be discussed:

- Term pruning.
- Term Weighting.
- stemming techniques.
- Normalization.

Then we will discuss:

- Effect of clustering algorithms.
- Effect of distance functions.

Many symbols were used in experiments setup for preprocessing combinations, as depicted below in Table 5.1.

Table 5.1 Symbols used in experiments and their description

symbol	description
Boolean	Indicating presence (1) or absence (0) of a word
wc	Output word counts
wc-tf	Apply *TF* transformation on word count
wc-tf-idf	Apply *TFIDF* transformation on word count
wc-norm	Apply document normalization on word count
wc-minFreq3	Apply term pruning on word count that less than 3
wc-norm-minFreq3	Apply normalization and term pruning on word count that less than 3
wc-tfidf-norm-minFreq3	Apply *TFIDF* and normalization on word count that less than 3
wc-norm-minFreq5	Apply normalization and term pruning on word count that less than 5
wc- tfidf-norm -minFreq5	Apply *TFIDF* and normalization on word count that less than 5

5.1 Analysis of Term Pruning Impact

Term pruning is applied for preprocessing in String to Word Vector options by setting the minimum term frequency in the document. In default state the minTermFreq =1, that means no term pruning are applied and all words are contained in dataset. We increased term frequency in many counts (3,5,7 and 9) to investigate the impact of term pruning in clustering process. The first dataset is CNN dataset, it is used in experiments with several values as followed.

Table 5.2: Precision, recall, and F-measure of using term pruning combining with term weighting and light stemming (CNN Dataset)

	minTermFreq=3	minTermFreq=5	minTermFreq=7	minTermFreq=9
precision	0.699	0.495	0.531	0.554
Recall	0.491	0.443	0.531	0.554
F-Measure	*0.577*	0.468	0.531	0.554

Table 5.3 : Precision, recall, and F-measure of using term pruning combining with term weighting and normalization (CNN Dataset)

	minTermFreq=3	minTermFreq=5	minTermFreq=7	minTermFreq=9
precision	0.614	0.625	0.591	0.620
Recall	0.492	0.436	0.536	0.498
F-Measure	0.546	0.514	*0.563*	0.552

Table 5.4 : Precision, recall, and F-measure of using term pruning combining with term weighting , normalization, and root-based stemming (Khoja) (CNN Dataset)

	minTermFreq=3	minTermFreq=5	minTermFreq=7	minTermFreq=9
precision	0.545	0.635	0.527	0.531
Recall	0.331	0.509	0.527	0.531
F-Measure	0.412	*0.565*	0.527	0.531

From tables 5.2, 5.3, and 5.4; results show that F-measure has the largest value for minimum term frequency at minTermFreq 3: 0.577, the largest measure is for minimum term frequency at minTermFreq 7: 0.563, and the last value is for minimum term frequency at minTermFreq 5: 0.565. From these results as shown the best value from these results is for minimum term frequency at 3. This gives indication that to use a small value for minimum term frequency to enhance results of text preprocessing as shown in Figure 5.1.

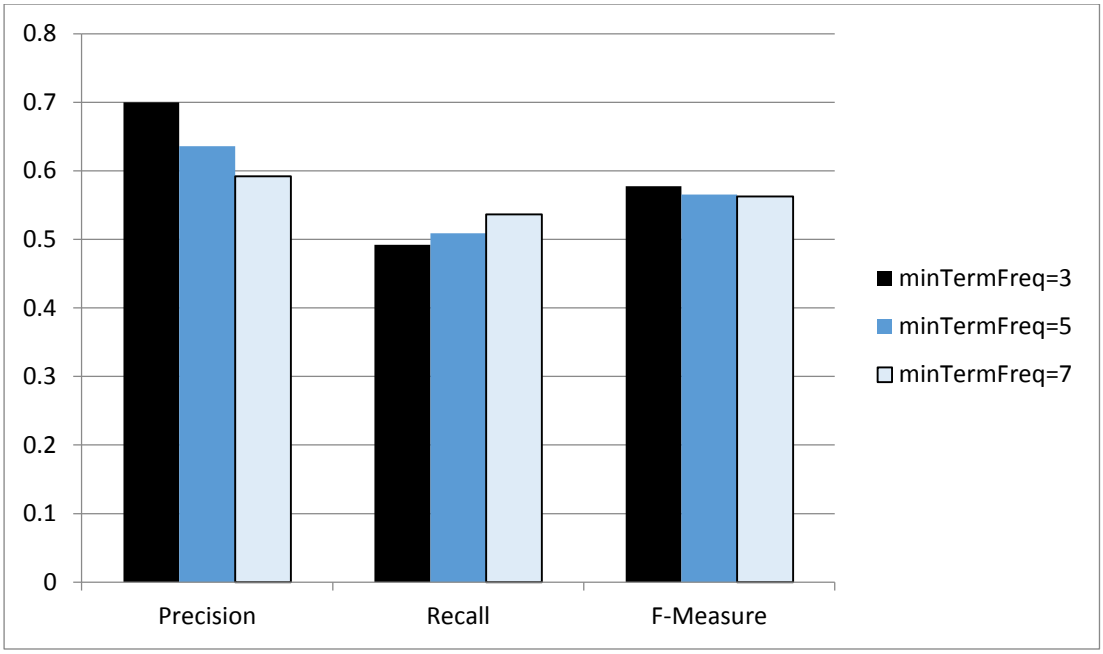

Figure 5.1: Evaluation of using term pruning with minTermFreq=3,5, and 7 (CNN Dataset)

For confirmation of term pruning impact, and the appropriate value for minimum term frequency, another dataset is used (BBC dataset) to show evaluation of term pruning with two values (3 and 5) .

Table 5.5 : Precision, recall, and F-measure of using term pruning combining with term weighting and normalization (BBC Dataset)

	minTermFreq=3	minTermFreq=5	minTermFreq=7	minTermFreq=9
precision	0.366	0.370	0.309	0.056
Recall	0.366	0.399	0.383	0.074
F-Measure	0.366	*0.384*	0.342	0.064

Table 5.6 : Precision, recall, and F-measure of using term pruning combining with root based stemming (khoja) (BBC Dataset)

	minTermFreq=3	minTermFreq=5	minTermFreq=7	minTermFreq=9
Precision	0.0195	0.052	0.321	0.321
Recall	0.019	0.052	0.330	0.420
F-Measure	0.019	*0.052*	0.325	*0.364*

Table 5.7 : Precision, recall, and F-measure of using term pruning combining with light stemming (BBC Dataset)

	minTermFreq=3	minTermFreq=5	minTermFreq=7	minTermFreq=9
Precision	0.786	0.739	0.321	0.321
Recall	0.692	0.684	0.345	0.345
F-Measure	*0.736*	0.710	0.332	0.332

From table 5.5, 5.6, and 5.7; results depicts that adjustment value of minimum term frequency at minTermFreq 3 gives the best evaluation for precision, recall, and F-measure, in comparison with other results. This also gives indication to use a small value for minimum term frequency to enhance results of text preprocessing. Figure 5.2 shows that minimum term frequency at minTermFreq 3 is the best value of evaluation for precision, recall, and F-measure.

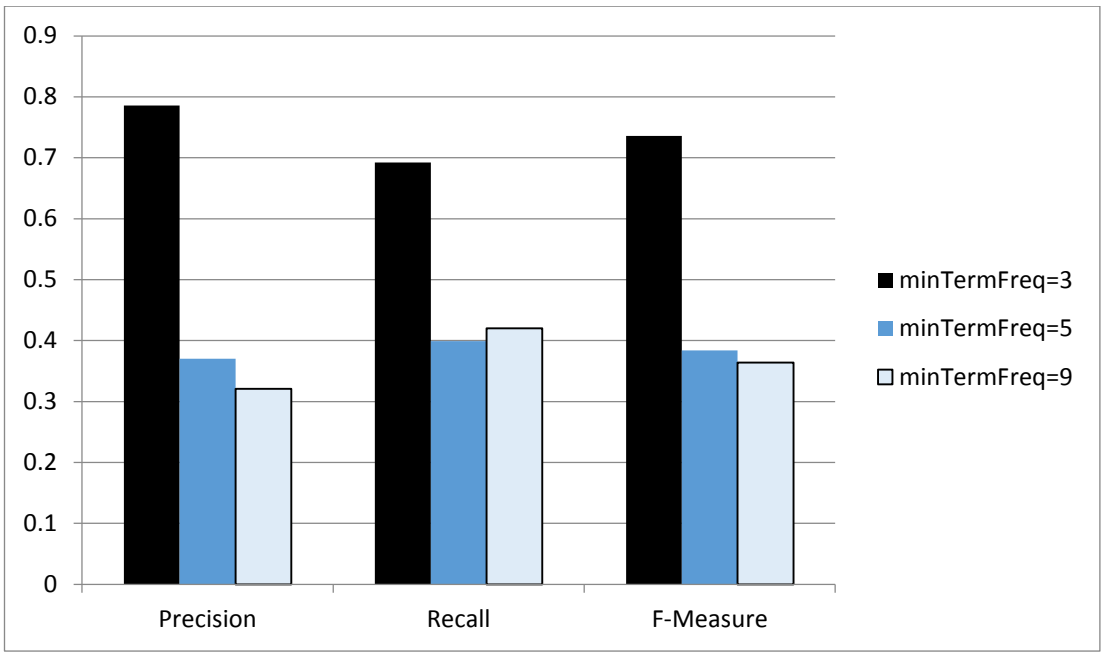

Figure 5.2: Evaluation of using term pruning with minTermFreq = 3, 5, and 9 (BBC Dataset)

5.2 Analysis of Term Weighting impact

Term Weighting aims to give higher weight to most discriminative terms. In this section, we will examine the impact of term weighting in document clustering. TF-IDF, which combines term frequency (TF) and inverse document frequency (IDF), and produce a composite weight for each term in each document, is used as term weighting. When using TF-IDF, evaluation is enhanced and results is better than without term weighting.

Table 5 .8: Precision, recall, and F-measure of using term weighting (TF-IDF) combining with term pruning (CNN Dataset)

	With (TF-IDF)	Without (TF-IDF)
Precision	0.614	0.148
Recall	0.492	0.148
F-Measure	*0.546*	0.148

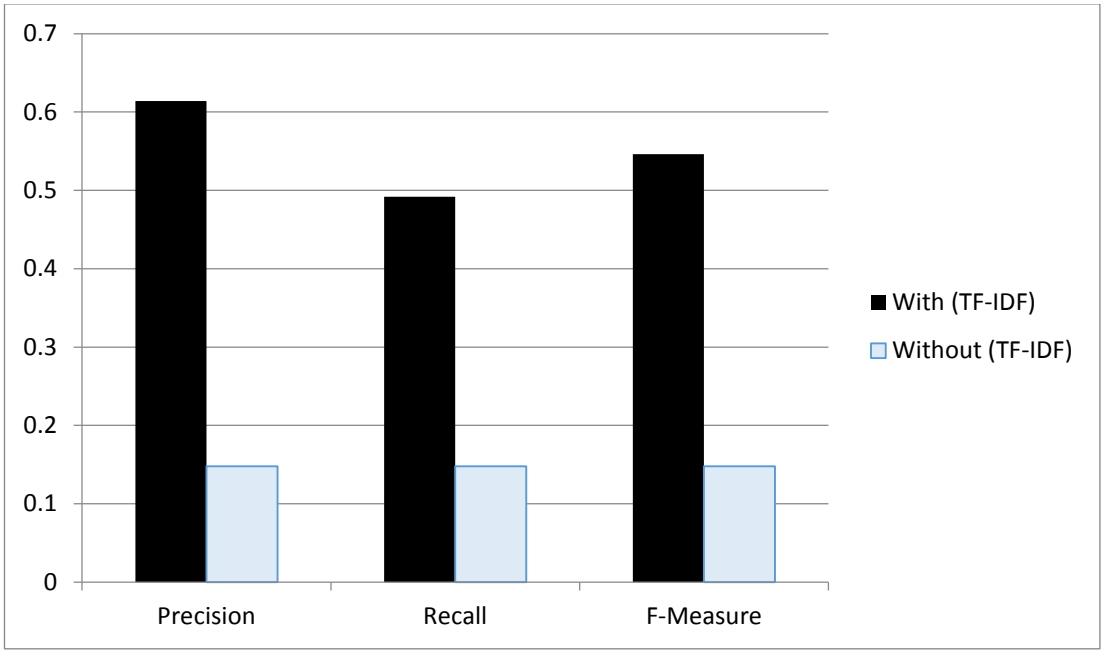

Figure 5.3: Evaluation of using term weighting (TF-IDF) combining with term pruning (CNN Dataset)

Table 5.8 shows precision, recall, and F-measure of using TF-IDF weighting with CNN dataset, which shows that the evaluation gets better

results when using TF-IDF weighting. Figure 5.3 depicts the evaluation graphically for using TF-IDF weighting combining with term pruning.

Table 5.9 : Precision, recall, and F-measure for term weighting (TF-IDF) combining with light stemming (CNN Dataset)

	With (TF-IDF)	Without (TF-IDF)
Precision	0.699	0.441
Recall	0.491	0.441
F-Measure	*0.577*	0.441

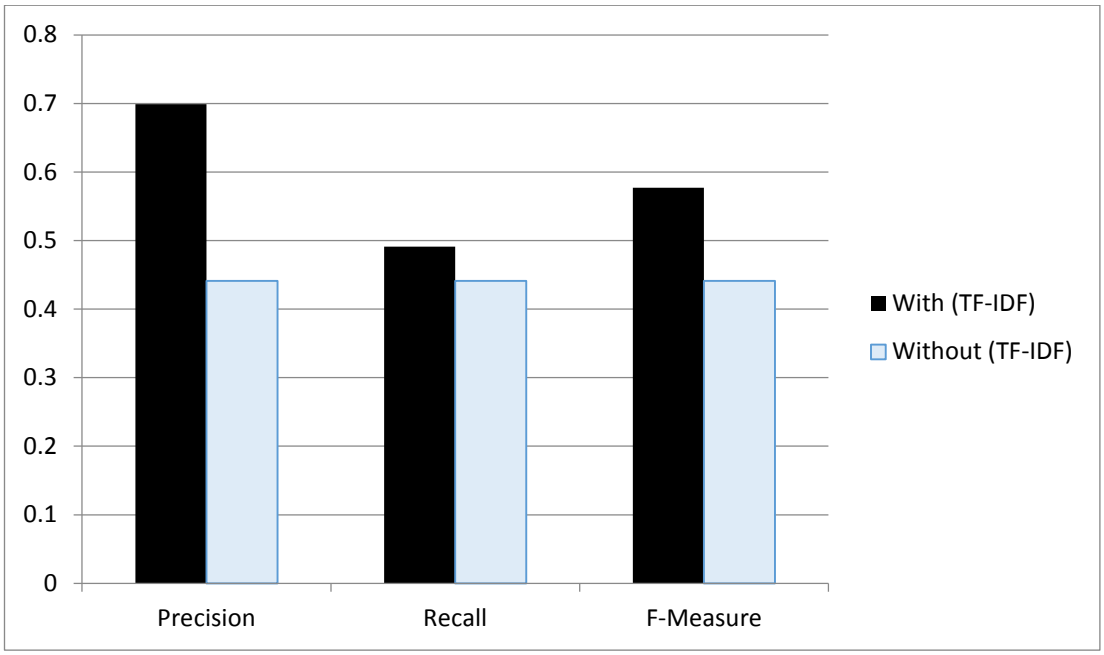

Figure 5.4: Evaluation of using term weighting (TF-IDF) combining with light stemming (CNN Dataset)

From Table 5.9 and Figure 5.4, results show that term weighting enhanced evaluation and give good results when using it in addition with light stemming.

Table 5 .10: precision, recall, and F-measure for term weighting (TF-IDF) combining with normalization and term pruning (CNN Dataset)

	With (TF-IDF)	Without (TF-IDF)
Precision	0.626	0.614
Recall	0.436	0.432
F-Measure	*0.514*	0.507

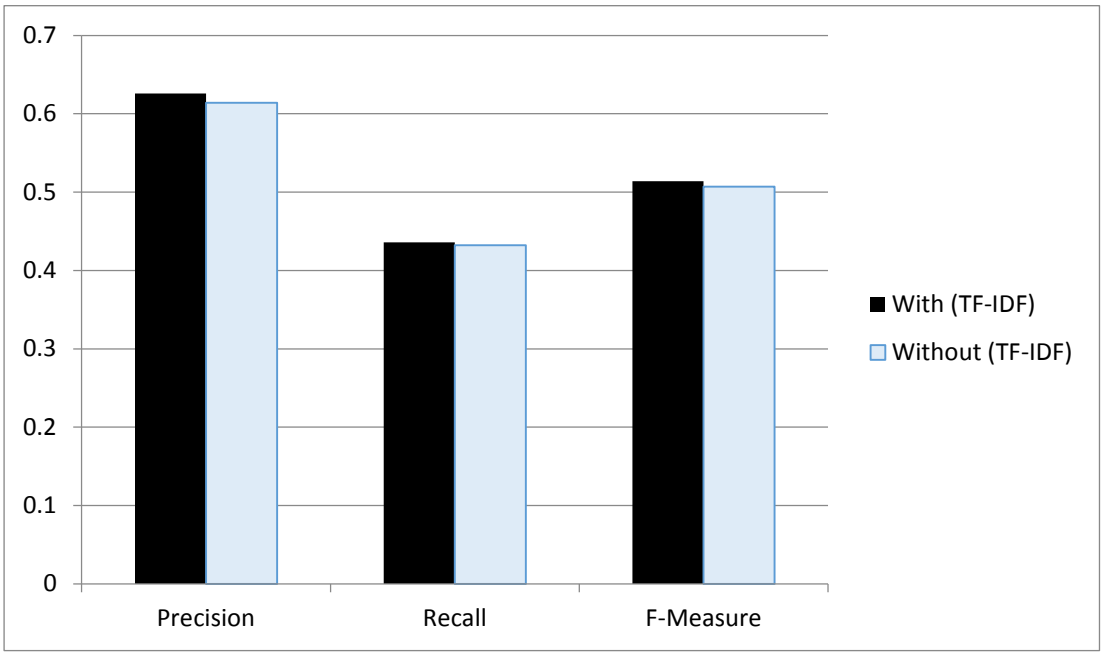

Figure 5.5: Evaluation of using term weighting (TF-IDF) combining with normalization and term pruning (CNN Dataset)

The result from Table 5.10 and Figure 5.5 show that term weighting enhanced evaluation slightly, as the evaluation is enhanced before using term pruning. Moreover, term weighting combining with normalization and term pruning makes more enhancements. Also, term weighting using (TF-IDF) has a positive evaluation effect and it enhances precision, recall, and F-measure to clustered documents.

In the other hand we experimented other dataset (BBC) and results is shown below in Tables 5.11,512, and 5.13.

Table 5.11 : Precision, recall, and F-measure of using term weighting (TF-IDF) combining with term pruning (BBC Dataset)

	With (TF-IDF)	Without (TF-IDF)
Precision	0.711	0.438
Recall	0.777	0.479
F-Measure	0.742	0.457

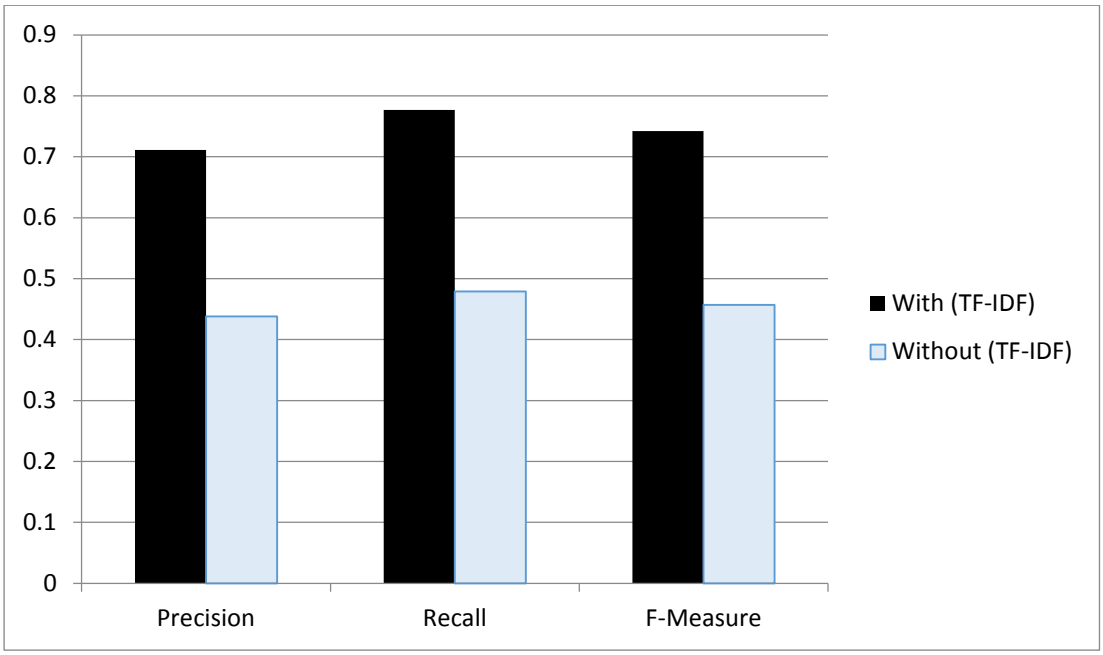

Figure 5.6: Evaluation of using term weighting (TF-IDF) combining with term pruning (BBC Dataset)

Table 5 .12: precision, recall, and F-measure for term weighting (TF-IDF) combining with normalization and term pruning (BBC Dataset)

	With (TF-IDF)	Without (TF-IDF)
Precision	0.158	0.132
Recall	0.182	0.152
F-Measure	0.169	0.141

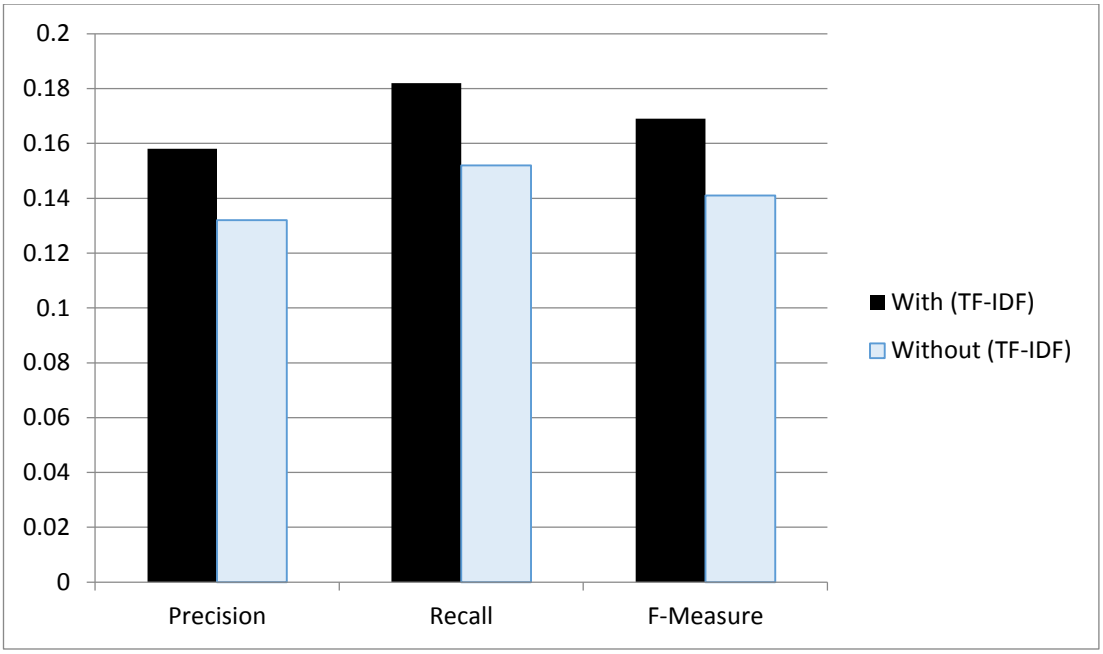

Figure 5.7: Evaluation of using term weighting (TF-IDF) combining with normalization and term pruning (BBC Dataset)

Table 5 .13: Precision, recall, and F-measure for term weighting (TF-IDF) combining with light stemming (BBC Dataset)

	With (TF-IDF)	Without (TF-IDF)
Precision	0.705	0.705
Recall	0.750	0.750
F-Measure	0.727	0.727

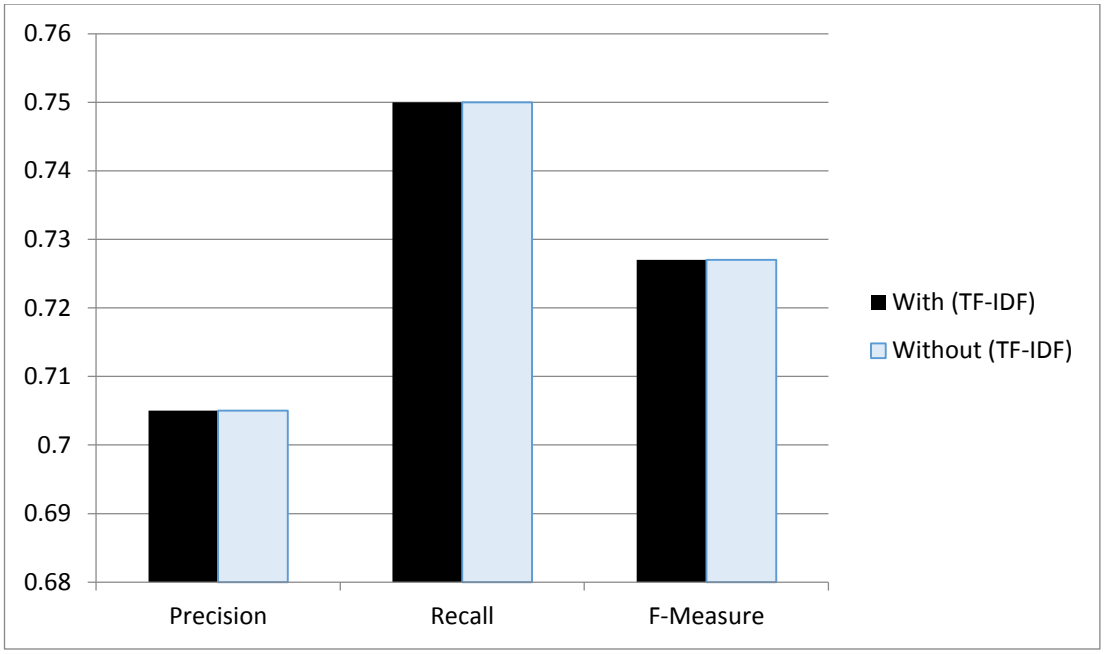

Figure 5.8: Evaluation of using term weighting (TF-IDF) combining with light stemming (CNN Dataset)

The result from Figures 5.6, and 5.7 show that term weighting enhanced evaluation, Moreover, term weighting combining with normalization and term pruning makes also enhancements. When term weighting combined with light stemming, no effect is achieved as shown in Figures 5.8.

From overall results, term weighting using (TF-IDF) gives a positive evaluation effect and it enhances precision, recall, and F-measure to clustered documents.

5.3 Analysis of Stemming Techniques Impact

In this section, we will evaluate stemming techniques in clustering of Arabic language documents and determine the most efficient in preprocessing of Arabic language, evaluation of applying three stemming techniques root-based Stemming, light Stemming, and without stemming (raw text).

Table 5.14: Comparing precision, recall, and F-measure for stemming techniques combining with term weighting (TF-IDF) (BBC Dataset)

	Light Stemming	Root-based Stemming (khoja)	Raw Text (without Stemming)
Precision	0.795	0.312	0.367
Recall	0.700	0.304	0.367
F-Measure	*0.745*	0.308	0.367

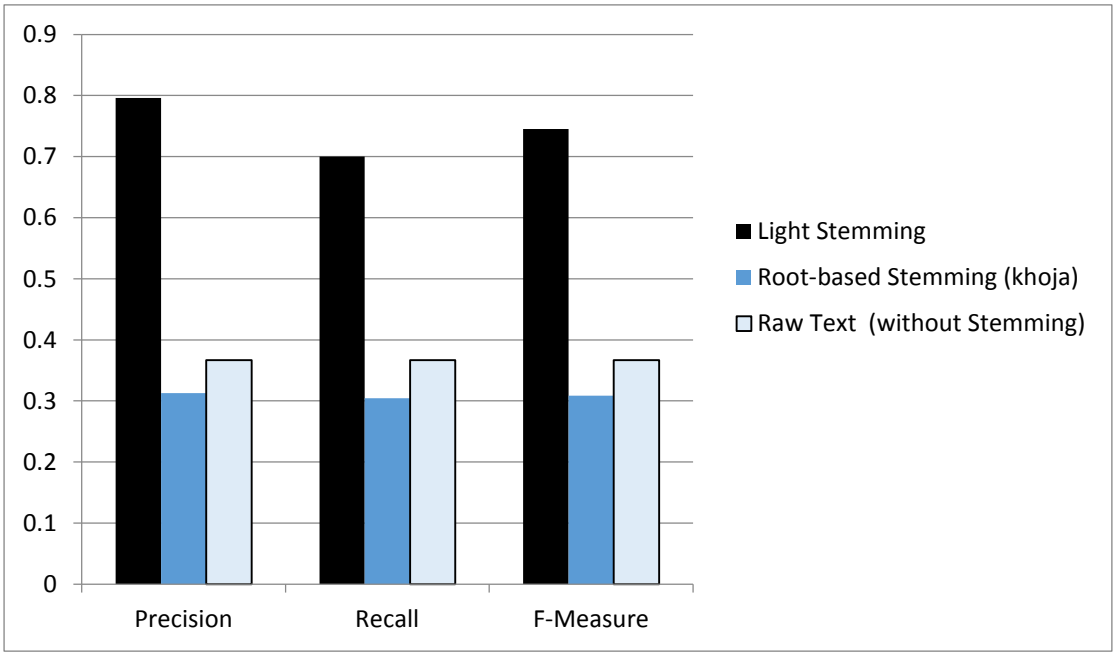

Figure 5.9: Comparing evaluation of using stemming techniques combining with term weighting (TF-IDF) (BBC Dataset)

In Table 5.14, evaluation measurements is shown for root based stemming (Khoja stemming is used in experiments), light stemming, and without stemming (raw text) for BBC dataset. Other preprocessing techniques are used combining with stemming, we perform test for best evaluation values stemming techniques. As shown in Figure 5.9, results of evaluation emphasize that light stemming has better evaluation than root-based stemming and raw text. In the other hand, root-based stemming (Khoja) enhanced the evaluation of clustering slightly, but its results didn't give the desired evaluation.

The results from other dataset ,CNN, (Table 5.15, and Figure 5.10) emphasize also light stemming gives best evaluation compared with other stemming techniques.

Table 5 .15: Comparing precision, recall, and F-measure for stemming techniques combining with term weighting (TF-IDF) (CNN Dataset)

	Light Stemming	Root-based Stemming (khoja)	Raw Text (without Stemming)
Precision	0.699	0.635	0.614
Recall	0.491	0.509	0.492
F-Measure	*0.577*	0.565	0.546

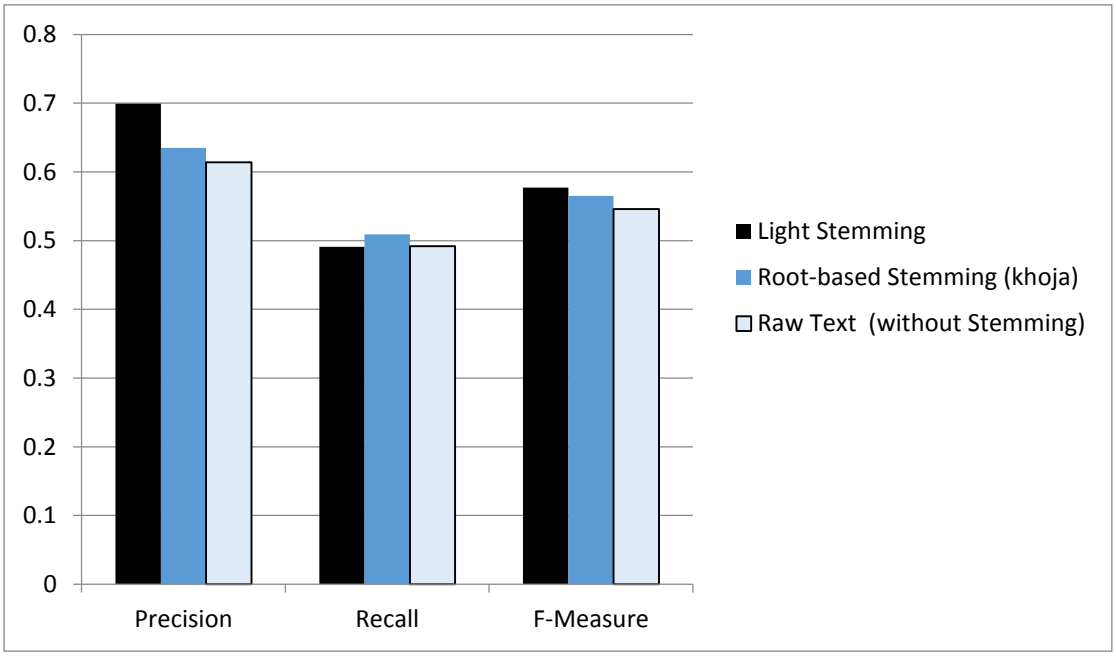

Figure 5.10: Comparing evaluation of using stemming techniques (CNN Dataset)

As observed from results of using BBC and CNN datasets evaluation, light stemming enhanced evaluation and gave the best result. therefore, light stemming in Arabic text clustering can be used to enhance clustering process, and this technique of morphological analysis is more appropriate than root-based stemming and raw text.

5.4 Analysis of Normalization Impact

Normalization is scaling data variables to be comparable. It is transforming tokens into a standard form. Normalization impact in document clustering will be investigated in this section.

Table 5 .16: Precision, recall, and F-measure for Normalization combining with term pruning (BBC Dataset)

	With Normalization	Without Normalization
Precision	0.523	0.479
Recall	0.477	0.438
F-Measure	0.499	0.458

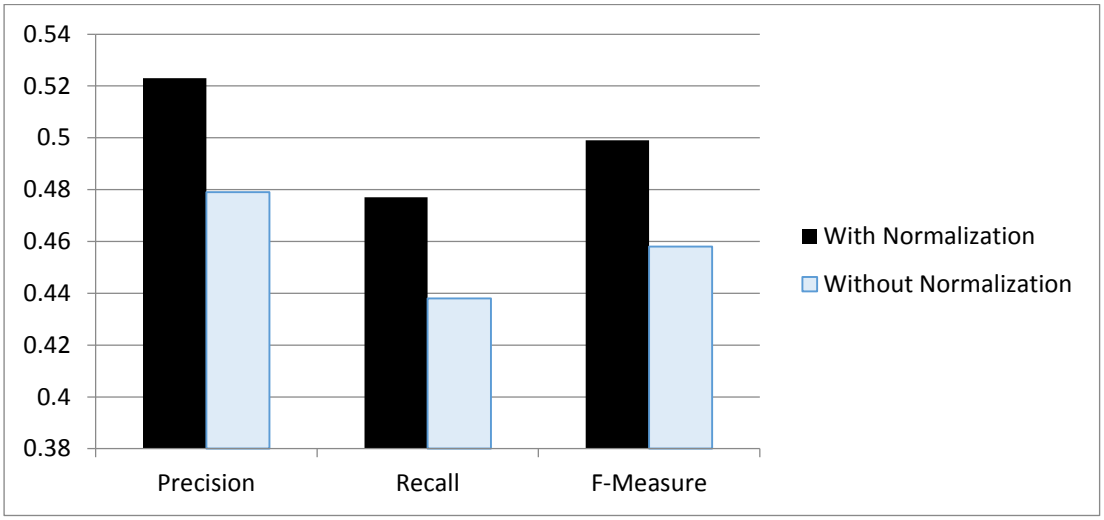

Figure 5. 11: Evaluation of using for Normalization combining with term pruning (BBC Dataset)

Evaluation results, as depicted from Table 5.16, and Figure 5.11, emphasize that normalization increases evaluation and enhance results slightly. Other experiments are applied for normalization combining with root-based stemming and term pruning, results is shown as followed.

Table 5.17: Precision, recall, and F-measure for Normalization combining with root-based stemming and term pruning (BBC Dataset)

	With Normalization	Without Normalization
Precision	0.289	0.019
Recall	0.263	0.019
F-Measure	0.275	0.019

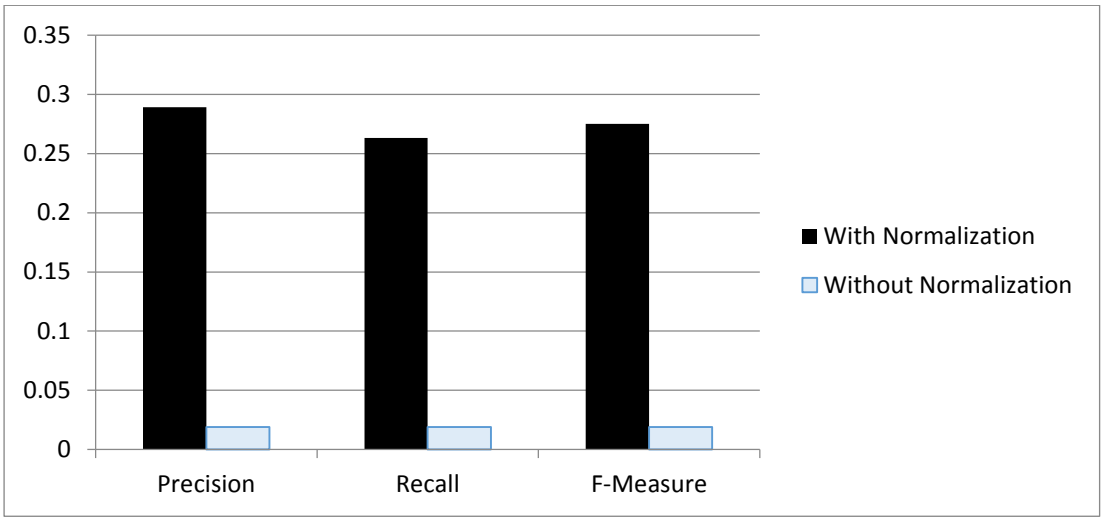

Figure 5.12: Evaluation of using normalization combining with root-based stemming and term pruning (BBC Dataset)

From Table 5.17 and Figure 5.12, evaluation shows a large impact of normalization to enhance results. In this experiment, Normalization is applied for BBC dataset combining with other two preprocessing techniques: root-based stemming, and term pruning.

For confirmation, another dataset , CNN, is used to examine effect of Normalization. Then we will discuss results for normalization and its impact depending on the two datasets.

Table 5 .18: Precision, recall, and F-measure for Normalization combining with term pruning (CNN Dataset)

	With Normalization	Without Normalization
Precision	0.525	0.462
Recall	0.471	0.414
F-Measure	0.497	0.437

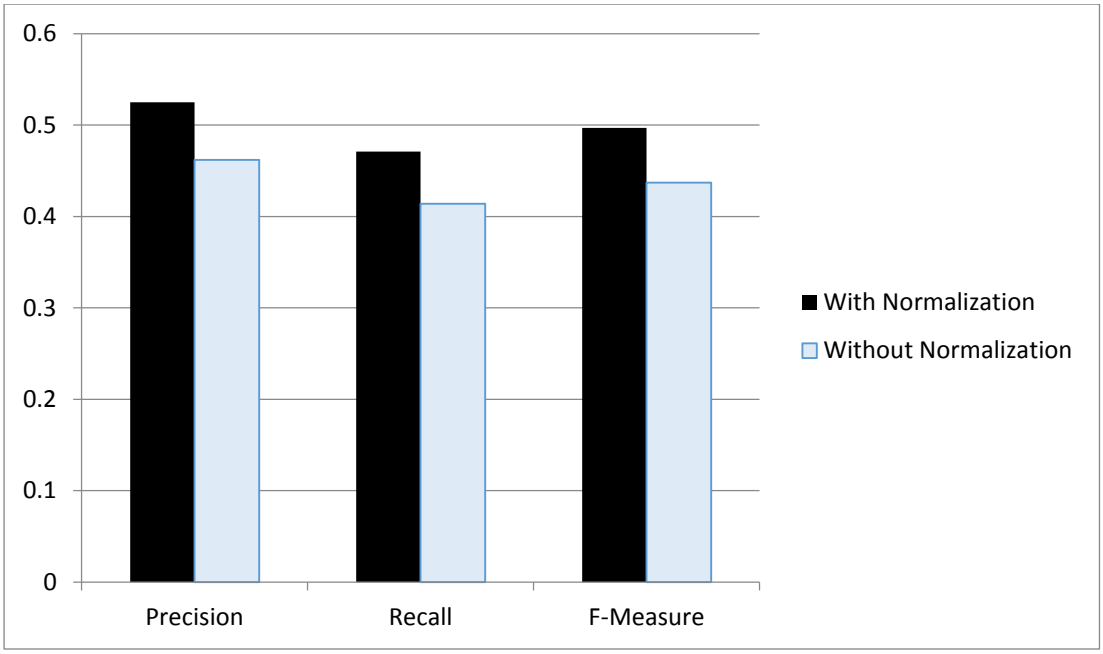

Figure 5.13: Evaluation of using for Normalization combining with term pruning (CNN Dataset)

Table 5.19: Precision, recall, and F-measure for Normalization combining with root-based stemming and term pruning (CNN Dataset)

	With Normalization	Without Normalization
Precision	0.423	0.423
Recall	0.472	0.472
F-Measure	0.446	0.446

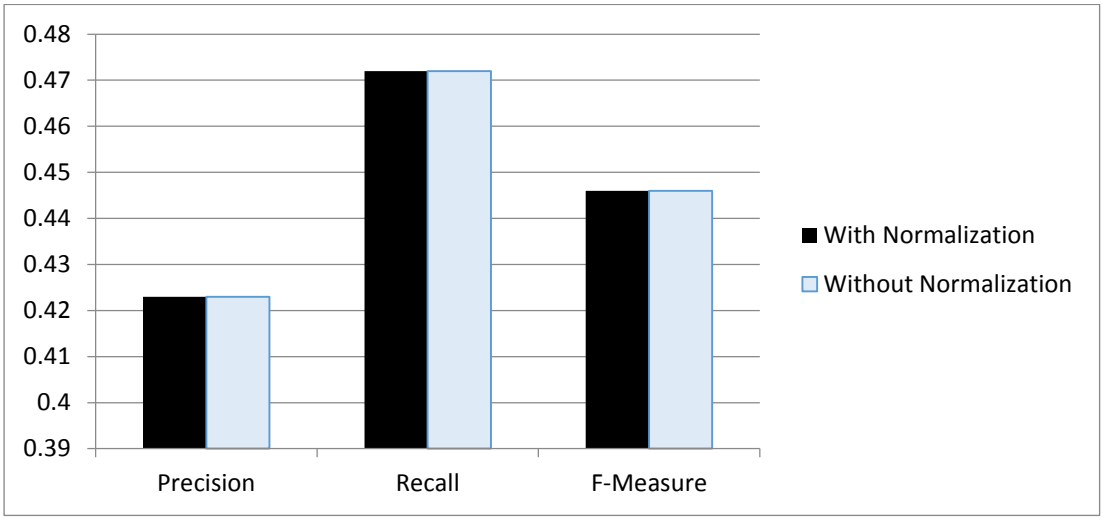

Figure 5.14: Evaluation of using normalization combining with root-based stemming and term pruning (CNN Dataset)

As shown in Table 5.18, and Figure 5.13; results emphasize the increase of precision, recall, and F-measure when normalization is applied for data. In the other hand Table 5.19, and Figure 5.14 emphasize that no effect of normalization when it combined with root-based stemming and term pruning, because the evaluation is enhanced with other combinations already.

From overall experiments of applying Normalization on data, results investigate that Normalization can enhance clustering process of documents and gives better evaluation than without Normalization.

5.5 Analysis of Using Clustering Algorithm

In this section we will compare evaluation using two clustering algorithms, the first is K-means, which has been mentioned in details, in chapter 3. The other algorithm is Expectation Maximization (EM) clustering algorithm; which is partition-based clustering algorithm, the same type as K-means algorithm.

Table 5 .20: Comparing precision, recall, and F-measure for using K-means and EM clustering algorithms

	EM Algorithm	K-Means Algorithm
precision	0.457	0.796
Recall	0.424	0.700
F-Measure	0.440	*0.745*

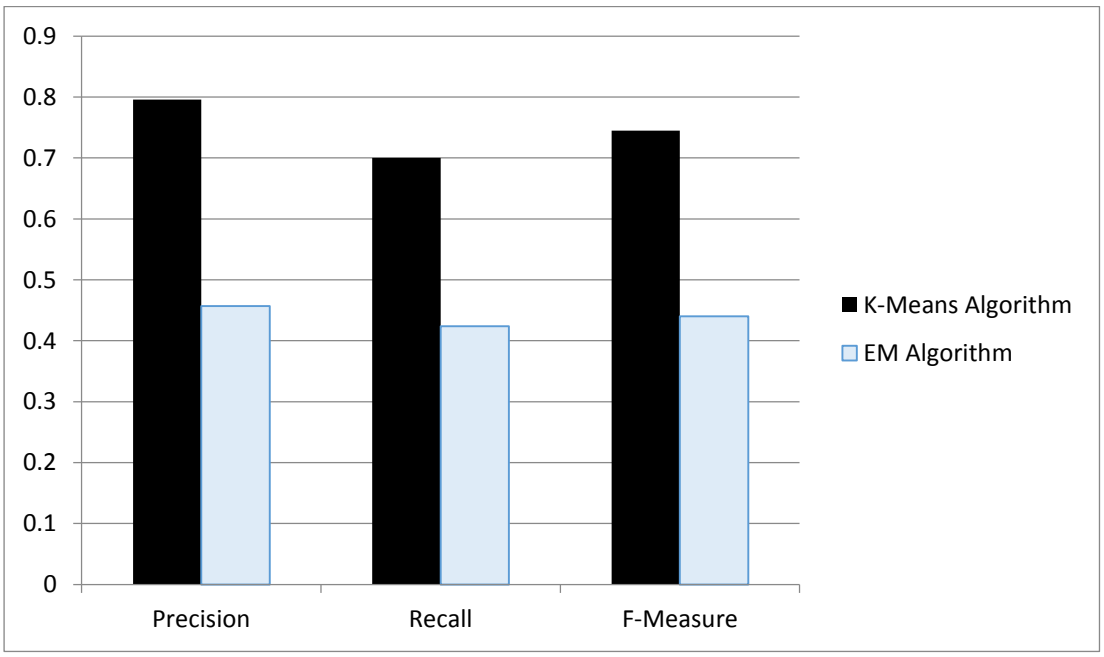

Figure 5.15: Comparing evaluation of using K-means and EM clustering algorithm

In this experiment, we used the data (BBC dataset) which gets the best value of evaluation in preprocessing techniques used previously. We evaluated these data by applying two clustering algorithms K-means and EM. Results in Figure 5.15 depict that K-means exceeds evaluation of EM algorithm.

5.6 Comparing of Using Distance Functions in Clustering Algorithm

Distance functions in k-means clustering technique play an important role. Different distance functions are provided to measure the distance between data objects [87].

In this section, we will compare two distance functions: Euclidean distance function, and Manhattan distance function, which will be used in K-Means algorithm.

- **Euclidean Distance Function**

Euclidean distance is ordinary distance between two points that one would measure with a ruler. It is the most commonly used distance function[88]. This distance is given by Pythagorean formula. The Euclidean distance between the points a and b is the length of the line segment connecting them (a, b) [89]. In the Euclidean plane, if a = (a_1, a_2) and b = (b_1, b_2) then the distance is given by:

$$D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \qquad (5.1)$$

This is equivalent to Pythagorean formula. Weakness of the basic Euclidean distance function is that if one of the input attributes has a relatively large range, then it can overpower the other attributes[89].

- **Manhattan Distance Function**

In Manhattan distance function the distance between two points is the sum of the absolute differences of their coordinates. The Manhattan distance, D_1 between two vectors a ,b in an *n*-dimensional real vector space with fixed Cartesian coordinate system, is the sum of the lengths of the projections of the line segment between the points onto the coordinate axis[88]. More formally,

$$D_1(a, b) = \|a-b\|^1 = \sum_1^n |a_i - b_i| \qquad (5.2)$$

where a = (a_1, $a_2 \ldots a_n$) and b = (b_1, $b_2 \ldots b_n$) are vectors.

Table 5 .21: Comparing precision, recall, and F-measure for using Euclidean distance function, and Manhattan distance function in K-means algorithm

	Manhattan Distance	Euclidean Distance
precision	0.382	0.796
Recall	0.351	0.700
F-Measure	0.366	*0.745*

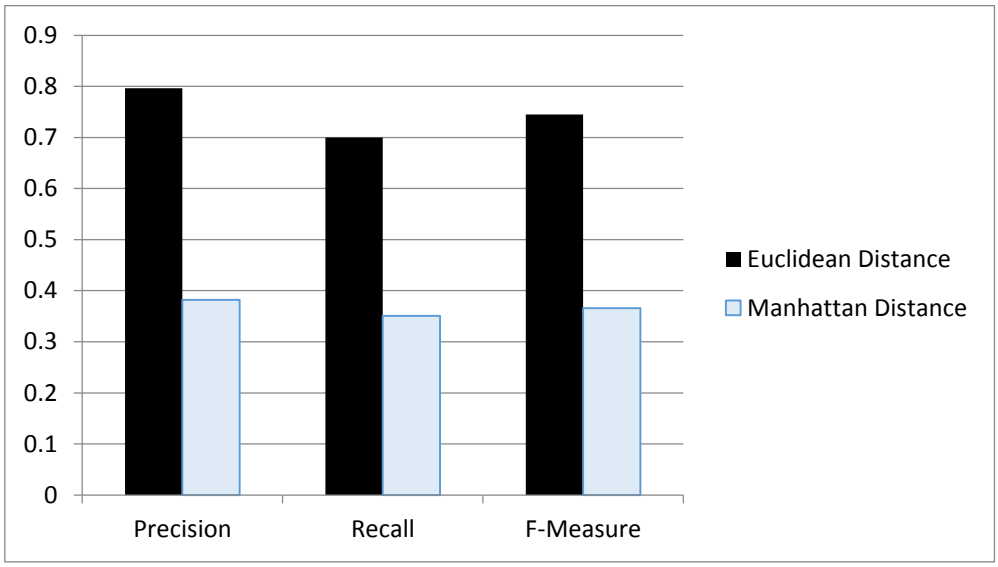

Figure 5.16: Comparing evaluation of using Euclidean distance function, and Manhattan distance function in K-means algorithm

In this experiment, BBC dataset with best preprocessing combinations, is used. This data gave the best evaluation in Euclidean distance, but evaluation fall back when using Manhattan distance. Therefore, Euclidean distance is more efficient in clustering algorithm for Arabic text clustering. These results are shown in Figure 5.16, and Table 5.21.

5.7 Summary

From the comprehensive results of using the BBC and CNN datasets, the observation of evaluation using precision, recall, and F-measure of applying minimum term frequency at minTermFreq 3 is the best value of evaluation. For applying term weighting using (TF-IDF), it affects in evaluation positively. Light stemming in Arabic text preprocessing can improve clustering process, and this technique of morphological analysis is more

appropriate than root based stemming and raw text. Performing Normalization on data, can enhance clustering process of documents and gives better evaluation than without Normalization. Results of using clustering algorithm show that K-means exceed evaluation of EM algorithm. In the other hand using Euclidean distance is more efficient in clustering algorithm for Arabic text clustering.

CHAPTER 6: CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this research we applied text preprocessing techniques to Arabic documents, then we achieve best combinations of these techniques when perform clustering algorithm. Experiments were applied to large corpora includes BBC corpus contains 1,860,786 (1.8M) words and 106,733 district keywords after stopwords removal, and CNN corpus contains 2,241,348 (2.2M) words and 144,460 district keywords after stopwords removal. Although complexity of Arabic language, we implemented analysis of using preprocessing techniques and investigated the impact of these techniques on Arabic text clustering. In our experiments, K-means clustering algorithm was used, we compared and examined this algorithm with other clustering algorithm Expectation Maximization (EM). The results confirmed that K-means is suitable for Arabic text clustering and gives better evaluation. On the other hand, comparison of distance measurements in clustering is performed for Euclidean distance and Manhattan distance, results investigated that Euclidean distance is more appropriate in Arabic text clustering.

From overall experiments, to enhance clustering process of Arabic documents many adjustments should be applied to give best evaluation results: In text preprocessing, applying term pruning with small value for minimum term frequency enhance results of text preprocessing. Results depicted that minimum term frequency at minTermFreq 3 is the best value of evaluation. Implementing term weighting **(**TF-IDF**)** also enhanced evaluation. In morphological analysis, light stemming is more appropriate than root-based stemming and raw text. Normalization also improves clustering process of Arabic documents, and evaluation is enhanced.

The best results of using these combinations induced measurements of evaluation for F-measure, precision, and recall : 0.745, 0.796, and 0.700 respectively, which give a good evaluation, and give an indication of using these combinations for Arabic document clustering to get suitable results.

6.2 Future Works

In Future work, some issues should be considered when enhancing clustering of Arabic documents, as follows:

1. use more datasets, and expand used corpus to contain more documents; to confirm the results and investigate our issues more broadly.
2. Reduce dimensionality of text data to reduce time of experiments and avoid out of memory problems.
3. In our experiments, we concentrate on partitioning based clustering algorithm, other clustering types can be compared with this algorithm and applied to data.
4. Expand application of clustering to documents contains other objects such as images, symbols, and figures.

References

[1] O. M. Al-Omari, "Evaluating the effect of stemming in clustering of Arabic documents," Academic Research International, vol. 1, p. 8, 2011.

[2] J. Han, M. Kamber, and J. Pei, Data mining: concepts and techniques, Morgan kaufmann, 2006.

[3] W. Fan, L. Wallace, S. Rich, and Z. Zhang, "Tapping the power of text mining," Communications of the ACM, vol. 49, pp. 76-82, 2006.

[4] D. A. Said, "Dimensionality Reduction Techniques for Enhancing Automatic Text Categorization," MSc. Thesis, Faculty of Engineering, Cairo University, 2007.

[5] J. Mariani, H. Uszkoreit, A. Zaenen, V. Zue, G. Varile, and A. Zampolli, "Survey of the State of the Art in Human Language Technology," ed: Citeseer, 1995.

[6] H. Chen, "Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms," Journal of the American Society for Information Science, vol. 46, pp. 194-216, 1995.

[7] Y. Yang, "Noise reduction in a statistical approach to text categorization," in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 256-263 , 1995.

[8] F. Sebastiani, "Machine learning in automated text categorization," ACM computing surveys (CSUR), vol. 34, pp. 1-47, 2002.

[9] M. A. Ismail and M. S. Kamel, "Multidimensional data clustering utilizing hybrid search strategies," Pattern Recognition, vol. 22, pp. 75-89, 1989.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," ACM computing surveys (CSUR), vol. 31, pp. 264-323, 1999.

[11] R. Duda, Hart, P., and Stork, D., Pattern Classification, (2nd Ed), Wiley Interscience, 2001.

[12] D. Hand, Mannila, H., and Smyth, P., Principles of Data Mining (Adaptive Computation and Machine Learning), MIT Press, Cambridge, MA, 2001.

[13] G. Salton, A. Singhal, M. Mitra, and C. Buckley, "Automatic text structuring and summarization," Information Processing & Management, vol. 33, pp. 193-207, 1997.

[14] U. Hahn and I. Mani, "The challenges of automatic summarization," Computer, vol. 33, pp. 29-36, 2000.

[15] S. J. Feldman R., The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data, Cambridge University Press, 2007.

[16] P. Jajoo, Document Clustering, Indian Institute of Technology, 2008.

[17] M. K. Saad and W. Ashour, "Arabic text classification using decision trees," presented at the Workshop on computer science and information technologies CSIT"2010, Moscow - Saint-Petersburg, Russia, 2010.

[18] "Arabic diacritics" - Wikipedia, the free encyclopedia, (2014, March), [Online]. Available: http://en.wikipedia.org/wiki/Arabic_diacritics.

[19] M. K. Saad and W. Ashour, "Arabic Morphological Tools for Text Mining," in The 6 th International Conference on Electrical and Computer Systems (EECS' 10), Lefke, North Cyprus, Nov25-26, 2010.

[20] V. K. Singh, N. Tiwari, and S. Garg, "Document Clustering using K-means, Heuristic K-means and Fuzzy C-means," in Computational Intelligence and Communication Networks (CICN), 2011 International Conference on, pp. 297-301, 2011.

[21] N. Sandhya, Y. S. Lalitha, V. Sowmya, K. Anuradha, and A. Govardhan, "Analysis of Stemming Algorithm for Text Clustering," International Journal of Computer Science, vol. 8.

[22] V. Tunali and T. T. Bilgin, "Examining the impact of stemming on clustering Turkish texts," in Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on, pp. 1-4, 2012.

[23] P. Han, D.-B. Wang, and Q.-G. Zhao, "The research on Chinese document clustering based on WEKA," in Machine Learning and Cybernetics (ICMLC), 2011 International Conference on. Vol. 4. IEEE, 2011. p. 1953-1957.10-13, Guilin, July 2011.

[24] M. S. Alkoffash, "Comparing between Arabic Text Clustering using K Means and K Mediods," International Journal of Computer Applications, vol. 51, 2012.

[25] S. H. Ghwanmeh, "Applying Clustering of hierarchical K-means-like Algorithm on Arabic Language," International Journal of Information Technology, vol. 3, 2005.

[26] M. Rafi, M. Maujood, M. M. Fazal, and S. M. Ali, "A comparison of two suffix tree-based document clustering algorithms," in Information and Emerging Technologies (ICIET), 2010 International Conference on, 2010, pp. 1-5.Karachi, 14-16 June 2010.

[27] A. A.-D. Abdelfatah A. Yahya, "Clustering Arabic Documents Using Frequent Itemset-based Hierarchical Clustering with an N-Grams," The 4th International Conference on Information Technology. Al-Zaytoonah University, Jordan. June 4th , 2009.

[28] M. H. Ahmed and S. Tiun, " k-means based algorithm for islamic document clustering," 07/2013; In proceeding of: IMAN 2013.

[29] H. Froud, A. Lachkar, and S. A. Ouatik, "Arabic text summarization based on latent semantic analysis to enhance arabic documents clustering," arXiv preprint arXiv:1302.1612, 2013.

[30] H. Froud, R. Benslimane, A. Lachkar, and S. A. Ouatik, "Stemming and similarity measures for Arabic Documents Clustering," in I/V Communications and Mobile Network (ISVC), 2010 5th International Symposium on, 2010, pp. 1-4. Rabat, Sept. 30 2010-Oct. 2010.

[31] W. M. A. Osama A. Ghanem, "Stemming Effectiveness in Clustering of Arabic Documents," International Journal of Computer Applications (0975 – 8887), vol. 49, 2012.

[32] C. D. Manning and P. Raghavan, "An Introduction to Information Retrieval Draft," Online edition. Cambridge University Press. -544 p, - 2009.

[33] A. K. Farahat and M. S. Kamel, "Enhancing document clustering using hybrid models for semantic similarity," in Proceedings of the eighth workshop on text mining at the tenth SIAM international conference on data mining. SIAM, Philadelphia, pp. 83-92, 2010.

[34] I. Yoo, "Semantic text mining and its application in biomedical domain," Drexel University, 2006.

[35] Yahoo!, (2014, March), [Online]. Available: " http://www.yahoo.com."

[36] ODP - Open Directory Project, (2014, March), [Online]. Available:", http://www.dmoz.org."

[37] Vivisimo Clustering Engine, (2012, April), [Online]."http://vivisimo.com/."

[38] Google News, (2014, March), [Online]. available:"http://news.google.com/."

[39] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," in Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 357-366, 2006.

[40] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in KDD workshop on text mining, pp. 525-526, 2000.

[41] C.-P. Wei, C.-S. Yang, H.-W. Hsiao, and T.-H. Cheng, "Combining preference-and content-based approaches for improving document clustering effectiveness," Information processing & management, vol. 42, pp. 350-372, 2006.

[42] N. Shah and S. Mahajan, "Document Clustering: A Detailed Review," International Journal of Applied Information Systems 4(5):30-38. Published by Foundation of Computer Science, New York, USA, October 2012.

[43] K. Mugunthadevi, S. Punitha, M. Punithavalli, and K. Mugunthadevi, "Survey on feature selection in document clustering," International Journal on Computer Science and Engineering, vol. 3, pp. 1240-1241, 2011.

[44] X. Cui, T.E. Potok, "Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm," Journal of Computer Sciences, vol. 5, pp. 27–33, 2005.

[45] G. K. Yi Peng, Zhengxin Chen, and Yong Shi, "Recent trends in Data Mining (DM): Document Clustering of DM Publications," Int'l Conference on Service Systems and Service Management, vol. 2, pp. 1653 – 1659, Oct. 2006.

[46] K. Mugunthadevi, S.C. Punitha, and M. Punithavalli, and Dr.M. Punithavalli, "Survey on Feature Selection in Document Clustering," Int'l Journal on Computer Science and Engineering (IJCSE), vol. 3, No. 3, pp. 1240-1244, Mar 2011.

[47] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," Pattern recognition, vol. 30, pp. 1109-1119, 1997.

[48] H. Wilson, B. Boots, and A. Millward, "A comparison of hierarchical and partitional clustering techniques for multispectral image classification," in Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International, pp. 1624-1626, 2002.

[49] U. Von Luxburg, "A tutorial on spectral clustering," Statistics and computing, vol. 17, pp. 395-416, 2007.

[50] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, pp. 788-791, 1999.

[51] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," Machine learning, vol. 42, pp. 143-175, 2001.

[52] M. Shameem, R. Ferdous. "An efficient K-Means Algorithm integrated with Jaccard Distance Measure for Document Clustering", Internet. AH-ICI 2009.First Asian Himalayas International Conference on, 2009.

[53] C.-R. Lin and M.-S. Chen, "Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, pp. 145-159, 2005.

[54] A. Huang, "Similarity Measures for Text Document Clustering," NZCSRSC 2008, April 2008, Christchurch, New Zealand, 2008.

[55] Z. Yao and C. Ze-wen, "Research on the construction and filter method of stop-word list in text Preprocessing," in Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on, pp. 217-221, 2011.

[56] A. Abdelali, Cowie, J. and Soliman, H. "Building a modern standard corpus," Workshop on Computational Modeling of Lexical Acquisition, In the Split Meeting, 2005.

[57] M. Tair and R. Baraka, " Design and Evaluation of a Parallel Classifier for Large-Scale Arabic Text," International Journal of Computer Applications 75(3):13-20, August 2013, New York, USA, 2013.

[58] R. Al-Shalabi, Kannan, G. and Gharaibeh, H. "Arabic text categorization using K-NN algorithm," The 4th International Multiconference on Computer and Information Technology (CSIT 2006) – Conference Proceedings, Amman, Jordan, 2006.

[59] L. Jing, Huang, H. and Shi, H. "Improved feature selection approach TFIDF in text mining," The 1st International Conference of machine learning and cybernetics – Conference Proceedings, Beijing, 2002.

[60] T.Gharib, M. Habib, and Z. Fayed, "Arabic Text Classification Using Support Vector Machines," The International Journal of Computers and Their Applications ISCA, vol.16, no.4, pp. 192-199, 2009.

[61] C. M. B. L. Larkey L., , "Light stemming for Arabic information retrieval," presented at the In Abdelhadi Soudi, Antal van den Bosch, and Günter Neumann, editors Arabic Computational Morphology: Knowledge-based and empirical method , volume 38 of Text, Speech and Language Tech-nology, Springer Verlag, 2007.

[62] F. Alotaiby, I. Alkharashi, and S. Foda, "Processing large Arabic text corpora: Preliminary analysis and results," in Proceedings of the second international conference on Arabic language resources and tools, pp. 78-82, 2009.

[63] M. K. Saad, "The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification," Master of Science, Computer Engineering, The Islamic University-Gaza, 2010.

[64] Weka Tutorial, (2014, March), [Online]. Available:", http://www.aboutdatamining.com/tutorials/weka-tutorial-2-data-scaling-via-normalization-and-standardization/."

[65] M. Syiam, Z. Fayed , and M. Habib , "An Intelligent System for Arabic Text Categorization," presented at the International Journal of Intelligent Computing and Information Systems IJICIS,vol. 6, no. 1, 2006.

[66] D. U. Smirnov I., "Overview of Stemming Algorithms," Mechanical Translation. DePaul University, Chicago, 2008.

[67] A. A. B. Sembok T., and Abu Bakar Z., "A Rule and Template Based Stemming Algorithm for Arabic Language," International Journal of Mathematical Models and Methods in Applied Sciences, Issue 5, Volume 5, pp. 974-981, 2011.

[68] Majdi, S., and Eric, A., "Comparative evaluation of Arabic language morphological analysers and stemmers," Presented at the Proceedings of COLING 2008 22nd International Conference on Comptational Linguistics, Manchester, UK, 2008.

[69] Sawaf H, Zaplo J., and Ney H. "Statistical Classification Methods for Arabic News Articles", Presented at the Arabic Natural Language Processing Workshop; Toulonse, France, July 2001.

[70] D. A. Said, N. M. Wanas, N. M. Darwish, and N. Hegazy, "A study of text preprocessing tools for Arabic text categorization," in The Second International Conference on Arabic Language, pp. 230-236, Cairo, Egypt, 2009.

[71] S. Khoja and R. Garside, "Stemming arabic text," Lancaster, UK, Computing Department, Lancaster University, 1999.

[72] M. a. F. Aljlayl, O. "On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach", ACM Eleventh Conference on Information and Knowledge Management; 2002 November 340-347; Mclean, VA, USA, 2002.

[73] Duwairi R., Al-Refai M., Khasawneh N. "Stemming Versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization", 4th Int. Conf. on Innovations in Information Technology. IIT '07. pp. 446 - 450. Al Ain, UAE, 2007.

[74] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Light stemming for Arabic information retrieval," in Arabic computational morphology, ed: Springer, pp. 221-243 , 2007.

[75] E. W. Weisstein, "Zipf's law," MathWorld–A Wolfram Web Resource, Available: http://mathworld. wolfram. com/ZipfsLaw. html,(Date Last Accessed on Jul. 22, 2011), 2012.

[76] A. Hoonlor, "Sequential patterns and temporal patterns for text mining," Rensselaer Polytechnic Institute, 2011.

[77] Z. Qiu, C. Gurrin, A. Doherty, A. Smeaton., "Term Weighting Approaches for Mining Significant Locations from Personal Location Logs," presented at the Proceedingsin CIT(2010), 2010 IEEE 10th International Conference on, pages 20 –25, 2010, Georgia, USA.

[78] S. M. Srivastava A., "Text Mining: Classification, Clustering, and Applications," presented at the Chapman & Hall/CRC, ISBN: 1420059408, 2009.

[79] A. W. Saad M., "Arabic Text Classification Using Decision Trees," presented at the Workshop on computer science and information technologies CSIT‟2010, Moscow - Saint-Petersburg, Russia, 2010.

[80] M. Lan, C. Tan, H. Low, and S. Sung, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," presented at the Special interest tracks and posters of the 14thinternational conference on World Wide Web, Chiba, Japan, 2005.

[81] J. Sanger, The Text Mining handbook: advanced approaches in analyzing unstructured data, Cambridge University Press, 2007.

[82] F. E. Hall M, Holmes B, "The WEKA data mining software: an update", ACM SIGKDD Explorations Newsletter, Vol 11, No.1, pp. 10-18, 2009.

[83] M. Mahdavi,H. Abolhassani, "Harmony K-means algorithm for document clustering". Data Mining and Knowledge Discovery, vol. 18, pp. 370-391, 2009.

[84] D.-B. W. Pu Han, Qing-Guo Zhao. " The research on Chinese document clustering based on WEKA," proceedings of the 2011 International Conference on Machine Learning and Cybernetics, Guilin, 10-13 July, 2011.

[85] X.-B. Xue and Z.-H. Zhou, "Distributional features for text categorization," Knowledge and Data Engineering, IEEE Transactions on, vol. 21, pp. 428-442, 2009.

[86] V. Amala Bai and D. Manimegalai, "An analysis of document clustering algorithms," in Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on, pp. 402-406, Ramanathapuram, 2010.

[87] R. Loohach and K. Garg, "Effect of Distance Functions on K-means Clustering Algorithm," International Journal of Computer Applications, vol. 50, 2012.

[88] A. Moore, "The case for approximate Distance Transforms," University of Otago, Dunedin, Presented at SIRC 2002 – The 14th Annual Colloquium of the Spatial Information Research Centre, University of Otago, Dunedin, NewZealand, 2002.

[89] D. R. Wilson, and T. R. Martinez, "Improved heterogeneous distance functions", J Artif Intell Res, vol. 6, pp.1 -34, 1997.