Ministry of Higher Education

and Scientific Research

University of Diyala- College of Science

Department of Computer Science

Pattern Discovery for Text Mining Measured by Levenshtien Distance

A Thesis

Submitted to the Department of Computer Science\ College of Science\ University of Diyala in a Partial Fulfillment of the Requirements for the Degree of Master in Computer Science

By

Layla Abd Al.Hak Ismael

Supervised by

Prof. Naji Mutar Suhaib

September 2019

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

﴿هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَّرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ مَا خَلَقَ اللَّهُ ذَٰلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ 5﴾

صدق الله العظيم

سورة يونس

الآية (٥)

الإهداء

أهدي جهدي المتواضع هذا

إلى الذي سهر على تعليمي بتضحيات جسام.....إلى مدرستي الأولى في الحياة

أبي الغالي على قلبي أطال الله في عمره

إلى من كانت سندي في الشدائد..... وكانت دعواها لي دائما بالتوفيق

أمي الغالية جزاها الله عني خير الجزاء في الدارين

إلى من كانوا ملاذي وملجئي.....إلى من تذوقت معهم أجمل اللحظات

إلى من هم اقرب إليّ من روحي.....أخواتي

إلى من آنسني في دراستي وشاركني همومي.....زوجي العزيز

إلى الشموع التي ذابت في كبرياء.....لتنير كل خطوة في دربنا

فكانوا رسلاً للعلم والأخلاق .....أساتذتي

ليلى عبد الحق إسماعيل

ACKNOWLEDGMENTS

First and foremost, I would like to thank to Allah SWT for his bless and mercy who has guided me in finishing this thesis. Then I would like to thank my supervisor, ***prof. Naji Mutar Suhaib****, professor of computer science at Diyala University – collage of science, for the extraordinary exertion he applied, I would like to thank him for his profitable direction and backing through his supervision of this work. I am very fortunate to receive such great support from him.*

I would like to thank my father, mother and sisters who have continued to provide encouragement and assistance over the last two years of study.

At long last, there are no words enough to thank my husband for being strong and having faith in me constantly and his support to complete this master project.

Abstract

The textual based information amount is rapidly accumulating Which stored electronically on our computers or Web. Any computer (laptop or desktop) is capable of accommodating enormous data amounts owing to the improvements in the storage devices.

Texts are included in text dataset and this dataset are unstructured. These unstructured data can be handled by text mining. The complexity and the considerable number for these data uncover numerous new capabilities to the analysts. Therefore, this work presents an enhancement of extracting useful patterns from text documents in the field of text mining using Pattern Taxonomy Model (PTM) and Levenshtein Distance Algorithm (LDA).

There are various methods to handle text documents. In this thesis, text mining system was suggested to overcome the problems that have occurred in term-based method and phrase-based method. The proposed system based on the behavior of LDA algorithm and PTM for determining the best accuracy of the extracted patterns with a short time and to prove that pattern based method is the best solution for text mining without any problems in the information extracted from the text.

The strength of the two algorithms (PTM, LDA) are tested using threshold values from 1 to 10 to get 1% to 10% of information in the text. The proposed system used "Openosis opinion dataset" and "Reuters 50_50 dataset" which stored in a file of ".txt" or text document.

The results of this test obtained by comparing among values of four features which are (global probability, local probability, absolute support, relative support) for the text to get higher average accuracy.

The results of proposed system have been compared with other systems. The proposed system get (98.68%) average accuracy for Unigram grammar and (99.65%) average accuracy for Bigram grammar while a system that used the Levenshtein Edit Distance for automatic lemmatization for modern English achieved an accuracy of 96% for English language and the system that used the process of pattern evolving and pattern deploying get 62% of precision and 82% of recall. So, using LDA with PTM achieved a better results compared to other systems.

Lists of Contents

Lists of Figures

Lists of Tables

List of Algorithms

List of Abbreviations

CCAT	Criteria Cognitive Aptitude Test
CSV	Comma-Separated Values
HTML	Hyper Text Markup Language
IE	Information Extraction
IF	Intermediate Form
IPE	Individual Pattern Evaluation
IR	Information Retrieval
IT	Information Technology
JSON	Java Script Object Notation
KDD	Knowledge Discovery in Databases
LDA	Levenshtein Distance Algorithm
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NOSQL	Not Only Structured Query Language
PAT-tree	Patricia Tree
PTM	Pattern Taxonomy Model
RCV1	Reuters Corpus Volume1
SQL	Structured Query Language
SUP_a	Absolute Support
SUP_r	Relative Support

SVM	Support Vector Machine
TF	Term Frequency
T_H	Threshold
TM	Text Mining
TREC	Text Retrieval Conference
TXT	Text Document
XML	Extensible Markup Language

Chapter One

General Introduction

Chapter One

General Introduction

1.1 introduction

80% of all the information saved in governments, businesses, industries, and other organizations are storing in the form of text. Texts are the most popular means of formally information exchanging. There is a requirement in our modernistic life to have a tool of business intelligence that is capable of extracting information as quickly as possible from texts. [1].

Data mining term is also called Knowledge mining that is the significant extracting of inherent, formerly unknown and possibly beneficial information from data in the dataset. It has many techniques such as Decision tree classifier, Neural network, Genetic algorithm, Rule extraction [2].

Different applications like business managing and market analyzing can benefit from the utilization of extraction information from a huge data amount. The process of knowledge discovery can be represented as a non-trivial extracting of information from big databases, information which is presented in an implicit manner in the data, formerly unknown and possibly beneficial to users. Data mining is a fundamental stage in the knowledge discovery process in dataset. Based on the approaches of data mining, a considerable number of patterns are created[3].

The processes of finding the effective use and updating the patterns remain an open research problem. The developed model of knowledge discovery satisfies this issue and is capable of applying the patterns in the domain of text mining [4].

The techniques of text mining are very useful to users for finding the desired knowledge from a massive data amount. It is extremely significant to retrieve efficient and relevant information for the users. Term-based approaches were used to provide these requirements. But these approaches have several drawbacks like polysemy and synonymy. Polysemy refers to a word which has several meanings, while synonymy refers to the words which have the same meaning[5].

For overcoming these drawbacks, the phrase-based approaches were presented. But these developed approaches also had several drawbacks such as

1- Lower statistical properties to terms.

2- The occurrence frequency of the phrases is minimal compared with the keywords

3- A considerable number of redundancy and noisy phrases.

In the presence of these drawbacks, pattern-based approach have been proposed to get rid of the problems in the previous approaches [6].

1.2 Related Work

- Ning Zhong et al. (2012) [7], proposed a technique for discovering patterns to handle the issues of low frequency and misinterpretation of text mining. The experiments were conducted on Reuters Corpus Volume1(RCV1) data collection and Text Retrieval Conference (TREC) topics. This technique also used the pattern deploying and evolving processes for finding the obtained patterns in text documents.
- Dipti S. Charjan and Mukesh A. Pund (2013) [8], proposed a technique for patterns discovery that encompasses the pattern deploying and evolving processes for improving the efficiency of utilizing and updating discovered patterns to find interesting and pertinent information. The presented technique used a document of .txt as input and applies different algorithms for getting useful patterns.
- Bharate Laxman, and D. Sujatha (2013) [9], presented a technique to discover patterns and then compute the patterns specificities to evaluate the concept of weights as per their distribution in the obtained patterns. This technique works on updating patterns which show ambiguity that is a characteristic called pattern evolution. Patterns deploying and evolving are also used. The obtained results on the prototype application expose that the obtained result is useful in the text data mining field.
- Charushila Kadu et al. (2013) [10], presented a hybrid system is working on minimizing the dimension dataset and similarity constraints. A feature-based analysis is used to reduce the dimensional of the massive datasets. This system uses the evaluation of feature for reducing the high dimensionality of

text vector, and then identifies the term frequency, after that, these frequencies are weighted via utilizing the inverse document frequency technique. The documents weight is utilized in clustering. The system combines both the semantic and similarity constraints and combined the Individual Pattern Evaluation PTM (IPE).

- V. Aswini and S. K. Lavanya (2014) [11], presented a technique that utilized the pattern taxonomy model for discovering the patterns from a large data amount and seeking for important patterns. This technique includes the pattern evolving and deploying processes for improving the efficiency of utilizing and updating derived patterns to find important and relevant information. This technique get 62% of precision and 82% of recall as a result.
- Shivani D Gupta and B.P.Vasgi (2015) [12], the designed system that concentrates on the performing of a specific manner to deriving the pattern, in addition, to utilize them for retrieving the relevant text. The pattern deploying and evolving are two processes that are utilized for developing the efficiency of the obtained patterns. Then, the obtained patterns are utilized to search for important and relevant information. to develop the efficiency of utilizing and updating obtained patterns regarding the view of users, the testing on RCV1 data collection fulfills the users needed data from a document.
- Vaishali Pansare (2016) [13], proposed an approach that uses the association rule mining based on the AprioriAll algorithm for discovering frequent patterns in text documents. The input can take different formats of a text file. It uses Hash Tree structure to store candidate itemsets, and find frequent

patterns and itemsets within less time. This proposed approach finds a solution to the misinterpretation and low-frequency issue. The obtained results have demonstrated that the time of execution needed by the algorithm is minimal than the time needed by the other compared algorithms.

- S. R. Lomate (2016) [14], The proposed system introduced a technique for pattern discovery that consists of the pattern deploying and evolving processes, for improving the efficiency of utilizing and updating obtained patterns for finding the important and relevant information. The fundamental prototype of Pattern Taxonomy Model (PTM) using Advanced Apriori Algorithm which focuses on the problem of getting beneficial frequent patterns from the documents. This technique used substantial tests on RCV1 data collection shows that the obtained result provides a hopeful performance. A comparison between the Apriori algorithm and Advanced Apriori Algorithm is done for finding frequent patterns in pattern mining and a show advanced Apriori has better computation time for frequent patterns. Reduce computing time and rule count for frequent pattern generation.
- H. M. Mahedi Hasan et al. (2018) [15], presented a technique for key terms extraction that is depending on semantic relation. This approach is working on extracting a specified number of keywords from documents for identifying the main text content. The data set is collected from various sources like newspapers, books, journals, etcetera. To extract these keywords, several machine learning and statistical techniques have been utilized such as Logistic regression, support vector machine, word co-occurrences, and PAT-tree. This technique presented an approach of modified semantic relation with an accuracy of 77.6% precision and 84.3% recall to chosen data sets.

1.3 The Problem Statement

Lots of researches in the field of text mining have concentrated on improving effective mining algorithms to discover various patterns from larger text documents. Hence, finding interesting and useful patterns remains an open issue. Within this field, the techniques of data mining can be utilized for finding a variety of text patterns, like frequent itemsets, and sequential patterns. This thesis deals with three problems which are:

1- Getting an effective deal with the massive amount of text document.
2- Discovering the useful patterns from digital text documents, and utilizing these mined patterns to develop the performance of the system.
3- How to improve and evaluate the efficiency of the discovered pattern from text documents.

1.4 Aim of the Thesis

The aim of this thesis is to:

1- Extract interesting patterns from text document while these patterns contain accurate information about the document.
2- Reduce the time required for finding the patterns which describe the content of the document.
3- Increase the accuracy of information extracted from the text by using data mining techniques such as:
   a- Summarization.
   b- information extraction.
4- Proof that the discovery of a pattern is the best solution for text mining.

1.5 Outline of the Thesis

The other chapters in this thesis are as follows:

Chapter Two: Theoretical Background

This chapter gives the background and review of text mining and its techniques: (classification, clustering, retrieval information, extracting information, and summarization), the stemming, N-gram and Levenshtein algorithm

Chapter Three: The Proposed System

This chapter describes the proposed pattern discovery system with its design and implementation.

Chapter Four: Experiential Results and Tests

This chapter explains the results and evaluation that have been getting from the proposed system.

Chapter Five: Conclusions and Suggestions for Future Works

This chapter presents the conclusions of this work. Furthermore, it justify provides suggestions for future work.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

In the current chapter the basic theoretical aspects of pattern discovery system in text mining are presented, a brief introduction to text mining which can be utilized to finds the patterns from text. In addition, it presents the background for various necessary preprocessing issues and techniques that had been used in this thesis.

2.1 Data Mining

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information –information that can be used to increase revenue, cuts costs, or both. Data mining software is allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. As the data are available in the different formats so that the proper action to be taken. Not only to analyze these data but also take a good decision and maintain the data[16].

when the customer will require the data should be retrieved from the database and make the better decision. The important reason that attracted a great deal of attention in information technology the discovery of useful information from large collections of data industry towards field of "Data mining" is due to the perception of *"we are data rich but information poor"*. There is huge volume of data but we hardly able to turn them in to useful information and knowledge for managerial decision making in business [16].

With the massive volume of data stored in a dataset, files, and other warehouses, it is very necessary, to evolve a strong process for analyzing and interpreting such data and to extract the good knowledge which could assist in making a decision. The answer to the above is the utilization of the data mining process. This process is working on extracting the hidden predictive information from huge databases; it is a strong technology with large possibility to assist organizations to focus on the most substantial information in their data repositories [17].

2.2 Text Mining

Because the text is the generality normal form to store the information, it is believed that text mining represents a higher commercial probability than data mining. Actually, the recently existing studies referred that 80% of the company's information is included in text documents. But, the task of text mining is more complicated than data mining since it deals with texts which are inherently fuzzy and unstructured [18].

The process of text mining works on discovering useful knowledge in text documents. The main challenge is to acquire accurate knowledge in text documents for helping employers to get their needs. Several applications like business management and market analysis are capable of benefiting from the utilization of the knowledge and information extracted from a huge volume of data. Knowledge discovery can efficiently utilize and update discovered patterns and implement them in the text mining field. Therefore, Data mining is a fundamental stage in the knowledge discovery process in dataset, that indicates, the data mining is having the whole techniques of the process of knowledge discovery and giving modeling step which is the application of

techniques and algorithms for calculation of search pattern or models. So, it is very necessary to provide a proper model of text mining with relevant efficiency which is capable of retrieving the information that employers need [19].

Text mining is the same as data mining excepting the tools of data mining that are constructed to handle the structured data, while text mining can handle the sets of semi-structured or unstructured data like full-text documents, electronic mails, Hyper Text Markup Language (HTML) files etcetera [19].

Generally, text mining framework includes two distinct portions:

1-Text refining which transforms free form into an Intermediate Form (IF) text documents

2- knowledge distillation which produces knowledge or patterns from IF. There are two types of IF:

1- Structured IF like the relational data representation

2- semi-structured IF like the conceptual graph representation.

IF can be concept-based in which every entity refers to concepts or objects of interests in a particular field or document-based in which every entity refers to the document [20].

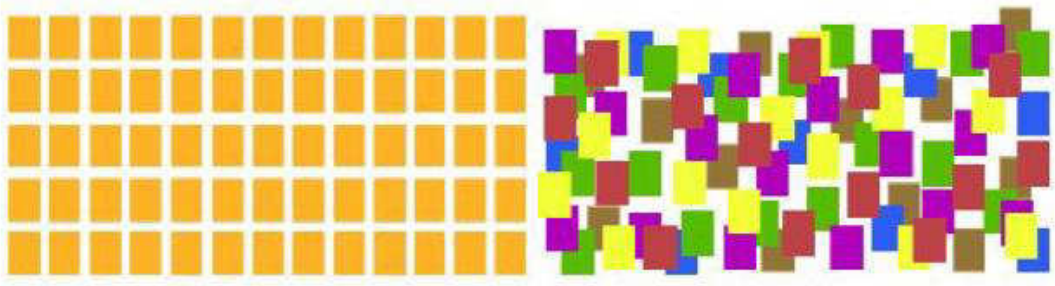
Unstructured data indicates to commonly computerized information which either doesn't have a model of data or can't easily be used via computer programs [20].

The Semi-structured data is information which doesn't exist in a relational database, and this may lead to having several organizational attributes which make it easy to analyze with several processes you can store them in relational

database (it could be very difficult to several kinds of semi-structured data), but the semi-structured are available to ease space, compute or clarity. Instance for semi-structured documents; Not Only Structured Query Language (NoSQL) databases, Comma-Separated Values (CSV) but Extensible Markup Language(XML) and Java Script Object Notation (JSON) documents [21].

Structured data is an organized structure, so, it is a term utilized for the actual data. The mostly utilized universal kind of structured data like Structured Query Language (SQL) and access are data sources. For instance; SQL, records (rows) and variables (columns) dependent information allows in select [22]. Figure (2.1) shows structured data versus unstructured data.

"80% of business-relevant information originates in Unstructured form, primarily text."



Structured data vs. Unstructured data

Figure (2.1): Structured Data vs. Unstructured Data [22].

Text Mining encompasses different functions, the essential ones are the search function, information extracting, classification, summarizing, setting priorities, clustering, information monitoring, and questions and responses [23].

2.3 The Text Mining Need

Text mining is beneficial to handle textual data. This type of data is unclear and very hard to manipulate, and unstructured, therefore, text mining is the most beneficial technique for exchanging information, while data mining is essentially implemented on business data [23].

Daily, Massive volumes of new data and information are produced out of academic, social and economic activities, with considerable possible societal and economic values. The data and text mining and analysis techniques are needed for exploiting these possibilities. The fundamental aim of this planning is to minimize the efforts needed to obtain information from a large number of text documents [24].

2.4 Text Mining Versus Data Mining

The text mining is different from data mining in the context of the data source . Essentially, the inputs in the text mining are unstructured files, but in data mining, the inputs are structured data [24]. Therefore, in text mining, the unstructured files are utilized to extract the patterns, whereas, in data mining, the structured data is utilized. Nowadays, the most existing business data is unstructured; although it may include facts, numbers, and dates in structured fields, typically, unstructured data is text (website text, articles, blog posts, etcetera). The existence of unstructured data leads to difficulty implement the activities of knowledge management utilizing conventional tools of business intelligence. The process of discovering the sources of knowledge which include unstructured data or text is named text mining [25].

Text mining is an outstanding manner for companies in the fields of business because most of the data in these locations are stored as text files. Therefore, the fundamental variation between text and data mining is that in text mining data is unstructured [25].

2.5 The Discovery of Patterns

The patterns are utilized as phrases or words which are extracted from the text documents. Lots of patterns can be discovered from the text documents, however, not all these patterns are useful. The patterns can be viewed as useful knowledge when evaluated in a specific way to be beneficial. It is a central process between association rule mining and inductive learning. It works on discovering patterns in labeled data which are descriptive. This process may face an issue when the discovered patterns are not useful (patterns are not suitable to be knowledge). So, the process of finding knowledge must have the ability to decide whether a pattern is useful enough to compose knowledge in the current context [26].

2.6 The Taxonomy of Patterns

The patterns are able to be constructed as a model of taxonomy-used knowledge discovery which is presented to apply the techniques of data mining in the applications of text mining. Knowledge Discovery in Databases (KDD) represents data mining term that works on finding the useful patterns from a database. Particularly, the process of KDD turns the low-level data into high-level knowledge [27].

Therefore, this process means data mining that extracts patterns from data. This thesis works on developing a model of knowledge discovery to efficiently utilize and update the discovered patterns and implement them into the text mining field [27].

2.7 Text Mining Methods

Text mining methods are developed on the basis of how text document is analyzed [18]. These methods are as follows:

1) Term-based method: The term in the document is utilized for identifying the content of the text. Each term is associated with the value known as weight. Here, the text document is analyzed on the basis of the term. The advantages of the term-based method include efficient computational performance as well as mature theories for term weighting. The fundamental disadvantage of this method is that the relation among words can't be reflected. The other issue is considering single words as features is the semantic ambiguity which can be categorized in [28]:

- **Polysemy:** means a word has multiple meanings.
- **Synonymy:** is multiple words having the same meaning.

2) Phrase-based method: The phrase is less ambiguous and carrying more semantics like information. Here, the document is analyzed on the phrase basis as phrases are more discriminative and less ambiguous than single terms. The probable causes for the discouraging performance involve [28]:

1) The phrases have low statistical properties to terms.

2) A low frequency of occurrence

3) Lots of noisy and redundant phrases are existing among them.

3) Concept-based method: This method works on finding a term which gives additional semantic meaning to document. Here, the term that contributes to sentence semantic is analyzed considering its importance at the sentence and the level of document. Usually, this model relies on the techniques of natural language processing. In order to remove ambiguity and noise, and optimize the representation, the feature selection is implemented to the query concepts [29].

4) Pattern-based method: In this method, the text documents are analyzed on the basis of the pattern. Patterns are item subsequences, sets, or substructures that appear in a data set with a frequency no less than a user specified threshold. The pattern utilized as a phrase or word which is extracted from the text document. To overcome the disadvantages of phrased based approaches, pattern mining based approaches have been proposed [30].

2.8 The Techniques of Text Mining

Several types of techniques are existing through which the text pattern mining and analysis are implemented. the main techniques are presented below.

2.8.1 Summarizing Text

Based on the number of documents, it is essential to produce summaries owing to a massive volume of information. Here, the document length will be minimized with no effects on the meaning of the document contents. A summary is producing from a single or a group of documents. A collection of documents is replacing by a summary [30].

For instance, the tools of summarization can extract the sentences that follow the keyword "conclusion", typically, after this phrase, the essential points of the document lie. Also, these tools can search for headings and subtopics for identifying the essential points of a document [31].

The process of automatic summarization is partitioned into several steps [31]:

Step1: *The step of preprocessing:* Obtain the structured representation of the original text.

Step2: *The step of processing:* Convert the structure of text into a structure of the summary.

Step3: *The step of generation:* Obtain the final summary from the structure of the summary.

In terms of the level in the linguistic space, the summarization methods can be divided into two groups:

- **Shallow techniques:** These techniques are restricting to the syntactic-level of representation and trying to extract prominent parts from the text in an appropriate manner.
- **Deeper techniques:** These techniques suppose a semantic-level of representation of the original text and include linguistic processing at some level.

2.8.2 The Classification Technique

The classification technique is a supervised technique that is depending on the group of input and output instances that are essentially utilized for training the model being utilized, for classifying the new documents [32].

This process finds the essential theme of the document via adding metadata and analyzing document. It finds the words counts and based on these counts, it will decide the document topic. In this technique, text documents are classified into predefined class label [32]. Figure (2.2) shows the classification technique.

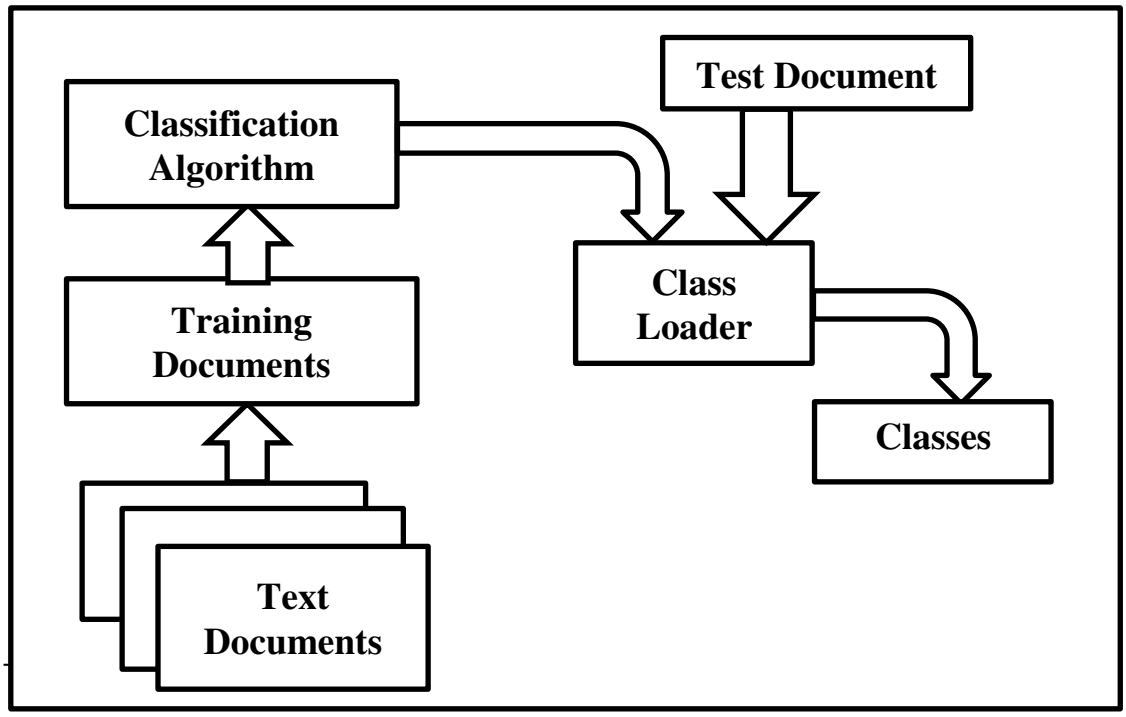


Figure (2.2): Classification [32].

There are several classification techniques can be implemented to classify the text such as:

1- Naïve Bayesian classification.

2- Nearest Neighbor classification.

2.8.3 The Clustering Technique

The text clustering technique is an unsupervised technique in which no input out patterns are pre-defined. This technique is depending on dividing the

similar text into the same cluster. Every cluster includes a number of documents. The clustering is looked better when the contents of the intra-cluster documents are more similar than the contents of inter-cluster documents [33]. The clustering technique is utilized to collect similar documents. This technique is worked on clustering the documents on the fly, while in the classification technique, the documents are clusters via the utilization of pre-defined topics. When the similarity is calculated, the algorithms of clustering should be implemented for generating classes list [33]. There are different categories of clustering: hierarchical clustering (Bottom-up and Top-Down), and partitioning clustering. Figure (2.3) shows the clustering technique.

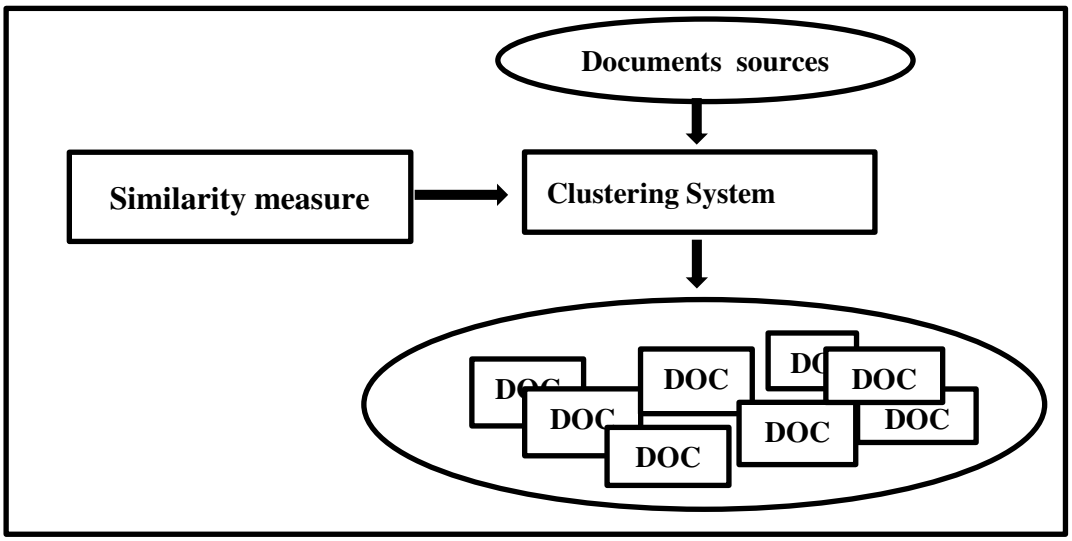


Figure (2.3): Clustering [33].

2.8.4 Information Extraction

The fundamental aim of information extraction techniques is extracting beneficial information from the text (see figure 2.4). It specifies the extraction of relationships, entities, and events from the unstructured or semi-structured text [34].

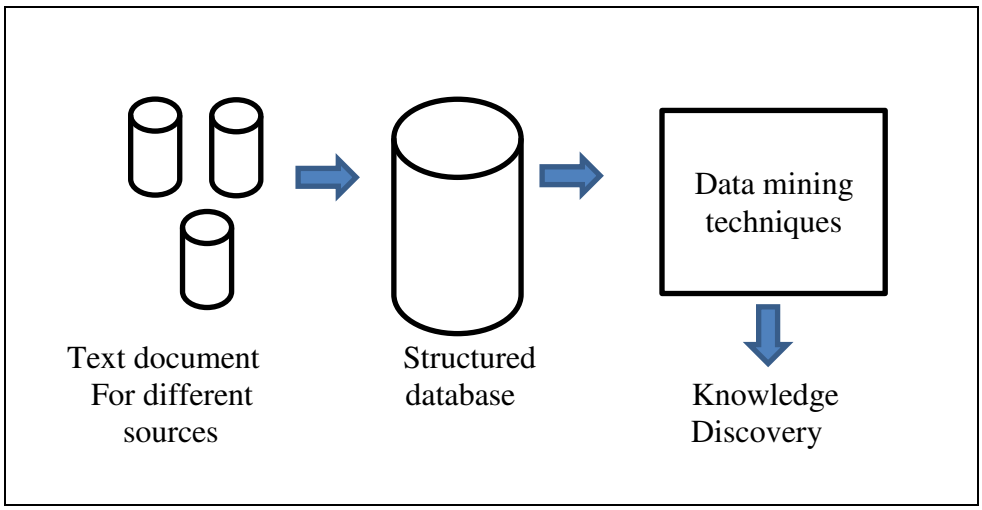


Figure (2.4): Information Extraction [34].

Information extraction is related to the semantic information extraction from the text in which the parts of the text are extracted and assigned a particular attribute to it. Mainly, Extracting Information identifies the words or feature terms from within a textual file. These feature terms are directly concurring with the domain [34].

2.8.5 Information Retrieval

The system of information retrieval (IR) is algorithms network that facilitates the searching for pertinent documents or data according to the user need. Besides providing the pertinent information for the user, it works on tracking the displayed data utility according to the user behavior[35]. The most famous IR systems are on-line library catalog systems, Google search engines, and online document management systems that recognize those an extension to document retrieval where the documents which are returned are processed for extracting the beneficial information interesting to the user [35]. IR steps are summarized in figure (2.5).

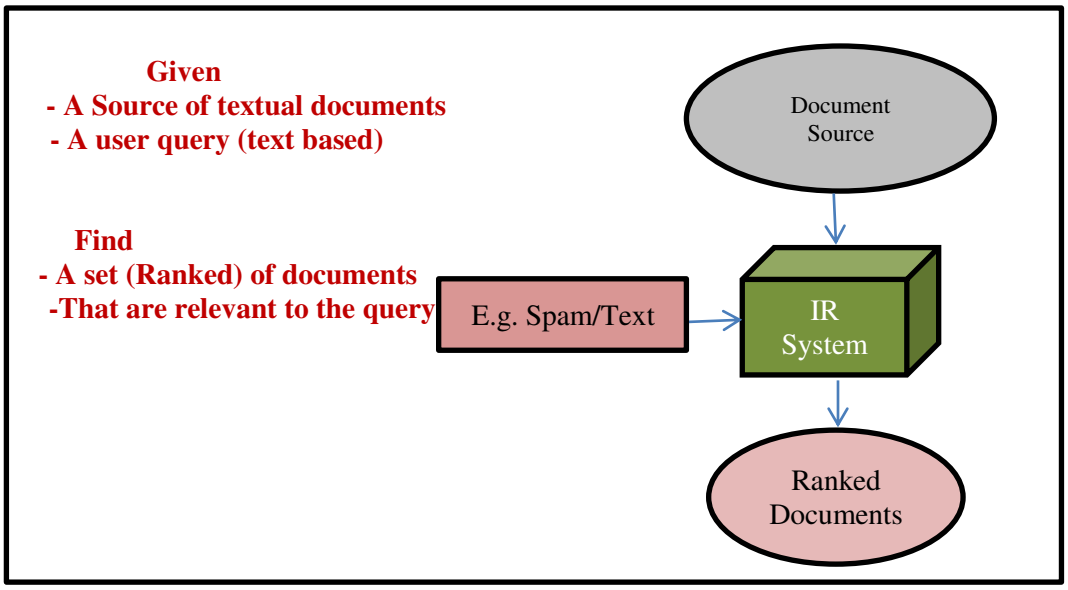


Figure (2.5): Information Retrieval [35].

After the stage of document retrieval, the stage of text summarizing which is focused on the posed user query, or the stage of information extraction are followed. In the wider sense, Information retrieval is dealing with the entire information processing range, from retrieving the information to retrieve the knowledge. It acquired maximized concerning with the growth of search engines and WWW [36].

Typically, the main issue of information retrieval is to locate relevant documents in a document collection depending on the query of the user, that is frequently some keywords describing an information requirement, though it could be an instance relevant document as well [36].

The set of documents relevant to a query be denoted as {Relevant}, and the set of documents retrieved be denoted as{Retrieved}.The set of documents that are both relevant and retrieved is denoted as {Relevant } ∩{Retrieved}[36], as shown in the Venn diagram of Figure (2.6).

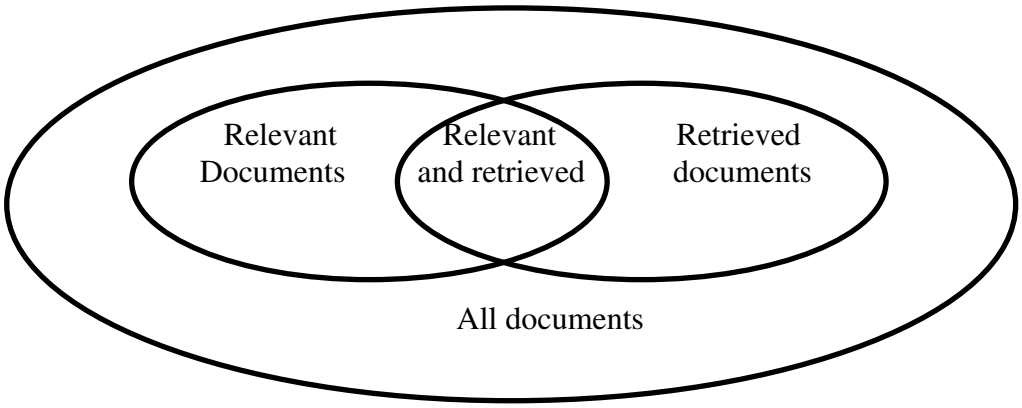


Figure (2.6): The Relation Between the Set of Relevant and the Set of Retrieved Documents.

2.8.6 The Visualization of Information

The visualization of information or visual text mining lays considerable textual sources in a visual map or hierarchy and supplies the abilities of browsing and plain searching. The visualization of information is beneficial if the user requires for narrowing down a wide range of documents and exploring related topics. The government is capable of using the visualization of information for identifying terrorist networks or for finding information about criminality, which may have been formerly considered unconnected. It could equip them, with a map of potential relationships among suspected activities in order that they can explore connections which they wouldn't have come up with on their own [37].

The aim of the visualization of information, the structure may include these steps [37]:

(1) The preparation of data: this means, specifying and obtaining the visualization of original data and forming original data space.

(2) The analyzing and extraction of data: this means, analyzing and extracting visualization data required from original data and forming visualization data space.

(3) Mapping of visualization: this means, employing a specific mapping algorithm for mapping visualization data space to visualization target.

2.8.7 Natural Language Processing (NLP)

NLP is a technology that concerns with natural language generation (NLG) and natural language understanding (NLU). The NLG utilizes some level of text underlying linguistic representation, for making assured that the text is grammatically generated fluent and correct. Most systems of NLG consist of:

1- Syntactic releaser for ensuring that grammatical rules like subject-verb agreement are followed.

2- The text planner for deciding how arranging paragraph, sentences and other portions coherently.

The generality famous application of NLG is the system of machine translation. This system works on analyzing texts from a source language into conceptual or grammatical representations and then generating corresponding texts in the target language [38].

The system of NLU calculates the representation of meaning, especially, restricting the domain discussion of computational linguistics. At least, NLU

includes one of the following parts; tokenizer, lexical analyzer, syntactic analyzer, and semantic analyzer. The tokenizer works on segmenting the sentence into a list of tokens. The token represents a special symbol or word like a question mark [38].

The lexical analyzer (morphological) works on tagging each word with its speech part. The complication attached to this analyzer when it is potential to tag a word with more than one speech part. The Syntactic analyzer assigns a parse tree or a syntactic structure, for a given natural language sentence. It specified, for example, how a sentence is separated into phrases, how the phrases are separated into sub-phrases, and all the way down to the actual structure of the words utilized [39]. In this thesis, some of data mining techniques such as summarization, information extraction are used.

2.8.8 A Comparison Among The Techniques of Text Mining

The text mining utilizes different techniques that take a significant role. These techniques vary among them. The classification is a supervised technique which utilizes the predefined set documents based on the contents. This technique finds the words counts and based on these counts decides the document topic. The clustering is utilized for finding essential structures in information and arranging them into related sub-groups for additional analysis and thesis [39]. Table (2.1) explains the comparison among text mining techniques**.**

Table (2.1): Comparison Among Text Mining Techniques [39]

Techniques	Characteristics	Tools
Information Retrieval	Retrievals valuable information from un structured text	Intelligent Miner , Text Analyst
Information Extraction	Extract information from structured database	Text Finder, Clear Forest Text
Summarization	Reduce length by keeping its main points and overall meaning as it is	Tropic Tracking Tool, Sentence Ext Tool
Categorization	Document based Categorization	Intelligent Miner
Cluster	Cluster collection of documents, Clustering, classification and analysis of text document	Carrot, Rapid Miner

.

2.9 Challenges of Text Mining Tasks

Although text mining has been studied for decades, there still exists several challenges and issues that should be addressed in this study, such as:

- **Performance**: Ideally, for each category, the algorithm is assumed to find everything relevant in the system (high recall) and only retrieve those into that category (high precision). The accuracy of the model depends largely on how the documents are represented as well as the distribution of the documents. For example, in document classification, the documents are usually in the form of vectors. Thus the vector space is usually quite large and sparse, which is the main cause of the "curse of dimensionality" phenomena.

performance has been the major measurement for almost all text mining tasks.

- **Scalability**: Efficiency is as crucial as performance for large-scale applications. To achieve better performance, usually, more training documents are preferable, which could in turn causes prohibitively long model training time for the algorithm. For example, in most scenarios of text classification, the training time is linear or quadratic to the number of training documents. e.g. the training time for support Vector Machine Classifiers (SVMs) is usually quadratic until a recent improvement which turns the training time to be linear. Some other simple classifiers, e.g., K-nearest neighbors classifier (KNN), logistic regression, their training cost is also decided by the number of categories in the training data.
- **Adaptivity**: An algorithm may perform well in one application (e.g., image categorization) but bad for another (e.g., text categorization). More commonly, a classifier may have different performance on different data sets in the same application. Thus building a universal algorithm that is both application-independent and dataset-independent becomes quite desirable [37].
  - **Customization(Personalization)**: Retrieving relevant documents based on user preferences has become a new research trend. For web users, the algorithms have to deal with documents with diverse content and users with diverse interests. Thus conventional algorithms (e.g., classifiers) that fix categories in advance obviously can't cater for all user interests [38].

2.10 Stemming

Stemming is a pre-processing step in text Mining applications and a very common requirement of NLP functions. At present-day, the word-stemming represents an essential feature provided by searching and indexing systems. Searching and indexing are a part of NLP systems, text mining applications, and IR systems. Usually, the stemming works on deleting any attached suffixes and prefixes (affixes) from index terms before the actual assignment of the term to the index. Actually, before applying any related algorithm, the text clustering, categorizing and summarizing need this transformation as part of the preprocessing as well [41].

While utilizing a stemmer, two points should be considered:

1- The morphological forms of a word that are assuming to have the same base meaning and it should be mapped to the same stem.

2- The words that don't have the same meaning, it should be preserved separate [42].

There are basically two types of stemming techniques, one is inflectional and other is derivational. Derivational stemming can create a new word from an existing word, sometimes by simply changing grammatical category (for example, changing a noun to a verb). The type of stemming we were able to implement is called Inflectional Stemming. A commonly used algorithms is the "Porters Algorithm" for stemming [43]. Figure (2.7) clarify kinds of stemming algorithms.

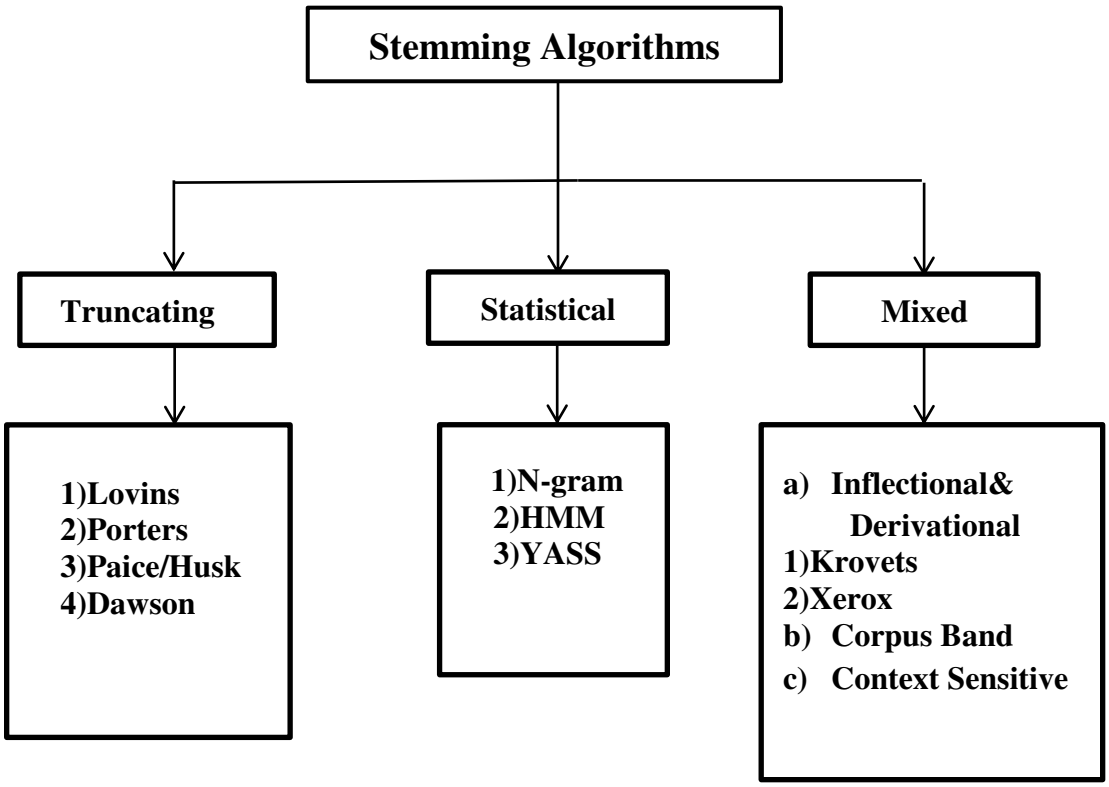


Figure (2.7): kinds of Stemming Algorithms [43].

2.11 Porter Stemmer

The algorithm of porter stemming is the most common stemming techniques presented in 1980. Porter algorithm is used to remove the stemming from the words and convert them into their base stem. It includes five stages, for every step, several rules are performed until the conditions are passed by one of the stages. When a rule is acceptable, the suffix stage should be ignored, and the subsequent step is applied[44]. Figure (2.8) illustrated the steps of porter stemming algorithm.

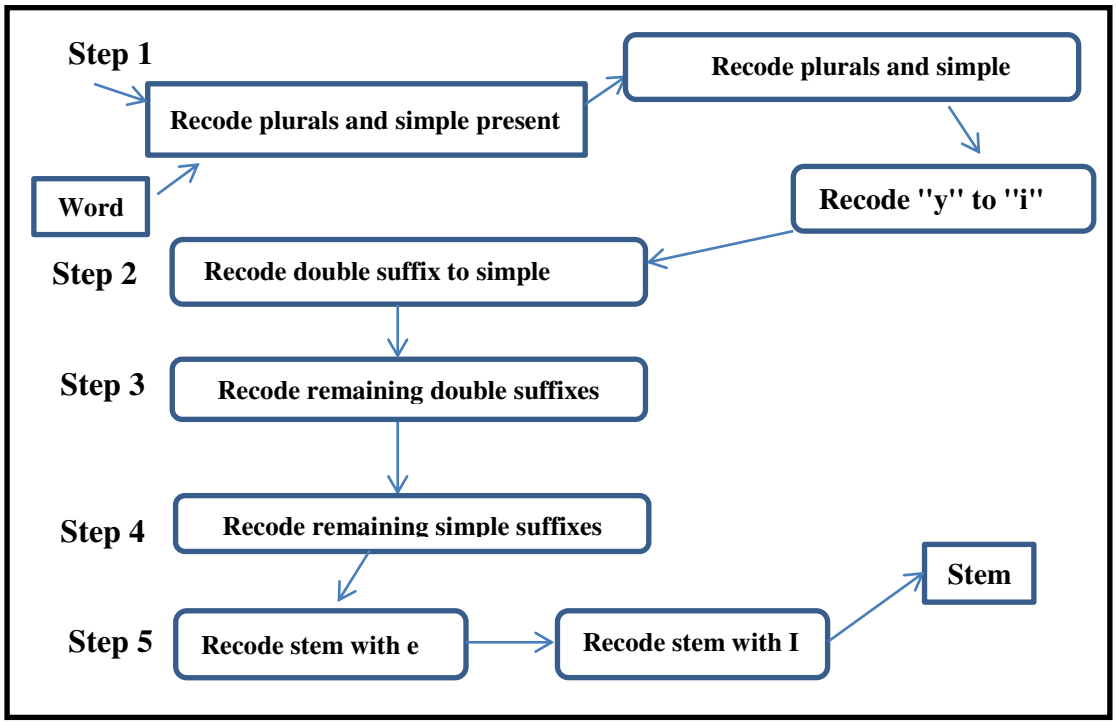


Figure (2.8): The steps of porter stemming algorithm [46].

The rule seems as the following [45]:

<condition> <suffix> → <new suffix>

For instance,

applied – applies– apply

printing – prints – printed– print

connection-connecting-connected– connect

In all situations, the whole words of the first instance will be handled as 'apply' and the whole words of the second instance will be handled as 'print'. The algorithm of porter has become the stemming English standard, and it gives a natural model for the processing to other languages. In several of those modern algorithms, the only relationship to the original is the utilization of a very limited suffix dictionary [45]. Algorithm (2.1) shows the standard algorithm of Porter algorithm.

Algorithm (2.1): Porter Algorithm[45]

Input: a set of Paragraphs.

Output: Paragraphs with only stems.

begin

Step1: Remove suffixes and plurals.

Step2: Turn the "y" letter to "i" letter in the existence of various vowels inside the stem

Step3: Do the mapping process of double suffixes to single ones: -ational, ation, etc.

Step4: Handle the suffixes such as; -ness, -full, etc.

Step5: Delete -ence, ant, etc.

Step6: Remove the last –e.

End

2.12 Edit Distance Algorithm (Levenshtein Distance)

The Edit Distance Algorithm (also known as Levenshtein Distance) on dictionary-based algorithm, which combines a string similarity and a most frequent in flexional suffixes model. Searching similar sequences of data is of great importance to many applications such as the gene similarity determination, speech recognition applications, database and/or Internet search engines, handwriting. Therefore, algorithms that can efficiently manipulate sequences of data (in terms of time and/or space) are highly desirable, even with modest approximation guarantees [46].

Algorithm (2.2) shows the standard algorithm of Levenshtein edit distance.

Algorithm (2.2): The Standard Algorithm of Levenshtein Edit Distance [46].

Input variables: char Text l[O..M-1], char Text2[O..N-1]
Output: the similarity between two words or sentences.
Step 1: declare: int d [O..M, O..N]
Step 2: for i from O to M
Step 3: d [i, O]: = i
Step 4: for j from O to N
Step 5: d [O, j]: =j
Step 6: for i from 1 to M
Step 7: for j from 1 to N
Step 8: if char of Text1 at (i-1)= char of Text2 at(j-1) then
Step 9: cost: = O
Step 10: else cost:=1
Step 11: end if
Step 12: d [i, j]:= Minimum (d[i-1,j] + 1, d [I, j- 1] + 1,d[i- l j- 1]+cost)
Step 13: end for (variable j)
Step 14: end for (variable i)
Step 15: return d[M, N];

Where:

d - Levenshtein matrix of size N+1, M+1, formed for Text1 and Text2 terms.

M ,N - are the two terms lengths respectively.

d [i, j] – (i, j) - represents an element in Levenshtein matrix d.

min – a function to compute the minimum of three variables.

cost - a variable which obtains either 0 or 1 values.

The distance *K* of Levenshtein is the minimum number of operations (deletion, insertion, and substitution) needed for changing the term into the another, as in the next equation:

$$K = d\,(M, N) \tag{2.1}$$

Levenshtein Distance is an easy dynamic programming algorithm which addresses the problem of sequence matching based on the notion of a primitive edit operation. By the term "primitive edit operation" we refer to the substitution of a symbol by another symbol, the deletion of a symbol and the insertion of a symbol [47].

It is clear that utilizing only the three mentioned operations of primitive edit, it is permanently potential for transforming an initial string A into a target string B (The two strings have the same alphabet). The distance of Levenshtein for these two strings is the minimum number of single-character substitutions, deletions, and insertions needed for transforming A into B [47].

Edit distance is the cost of unit operation which is needed for transforming a string to another one, where these two strings are becoming the same string. Unit operations can be categorized into four operations: deletion, insertion, transposition, and replacement. The deletions and insertions have the same costs, while the replacements have double of the insertion cost [48].

2.13 The measure of similarity among terms

The measure of similarity P is the quotient of a number of Levenshtein operations (after calculation of Lda) by the number of all Levenshtein operations in pessimistic case. This means, before the calculations of Lda will be completed but with the maximum possible number of Levenshtein operations well known [48]. The similarity measures *P* is calculated by the formula:

$$P=1-(\frac{K}{Kmax}) \; ; \; K_{max} = \max(N,M) \qquad (2.2)$$

$$K \geq 0 \; , \; M > 0 \; , N > 0 \; , \; P \epsilon \; (0,1)$$

where:

***K* max** – the length of the longest of analyzed two terms/text strings (i.e. pessimistic case

***K* –** is equal to the length of the longest term. Table (2.2) illustrates an example of the Levenshtein distance and the measure of similarity between two sentences.

Table (2.2) : An example of the Levenshtein distance and the measure of similarity between two sentences[48].

No.	Sentence1	Sentence2	K	P
1	My car isn't working	My bicycle isn't working	1	0.75
2	What did you do yesterday?	What have you done?	3	0.40
3	Tom is writing a letter	Tom is written letters	3	0.40

2.14 N-grams

An N-gram is a sub-sequence of n-items in any given sequence. In models of computational linguistics, N-gram is utilized generally in predicting words (in word-level N-gram) or predicting characters (in character-level N-gram) for different applications. Most applications in NLP and IR include extracting sequences of successive words (generally referred to N-grams) from large text documents. Extracted N-grams from text data can be utilized for different purposes. In machine translation, for instance, N-gram has utilized for constructing a model of statistical language [49].

The process of extracting N-grams from considerable text documents is a challenging issue especially if the word with longer sequences are considered. The number of distinct sequences increases as the data get bigger. With a large group of text documents, finding and quantifying the frequencies for every N-gram might need huge computational resources. Frequency of a term (TF) can be computed by using equation (2.3)[50]:

$$TF = N_k / N \quad (2.3)$$

Where

$\mathbf{N_k}$: is the ratio of the number of times a keyword K appears in a given document

N :is the total number of terms in that document.

N-gram have some features such as:

Absolute Support (Sup_a): absolute support can be computed by using equation(2.4):

$$sup_a(X)=|X| \quad (2.4)$$

where

|X|: is the number of paragraphs in which the term appeared.

Relative Support (Sup_r): relative support is calculated by Implement the equation (**2.5**) :

$$Sup_r(x)=\frac{|X|}{|PS(d)|} \quad (\mathbf{2.5})$$

Where

|PS(d)|: is the number of paragraphs in the document.

Global Probability: it can be computed by using equation (**2.6**):

Global probability = 1/ Number of occurrence of a term in the whole document (**2.6**)

Local probability: it can be computed by implement equation (**2.7**):

Local probability = 1 / Number of occurrence in paragraph (**2.7**)

Covering set: covering set can be computed by using the following equation:

Covering set= the paragraphs in which the term appeared (2.8)

N-gram approach extracts a set of phrases (depending on the value of n) rather than just a single phrase. For instance, the word "TEXT" would be composed of the N-grams (in character level N-gram) as follows:

bi-grams: _T, TE, EX, XT, T_

tri-grams: _TE, TEX, EXT, XT_, T_ _

quad-grams: _TEX, TEXT, EXT_, XT_ _, T_ _ _

and in word level N-gram for example:

bi-grams: San Francisco

tri-grams: The Three Musketeers (is a 3-gram)

quad-grams: She stood up slowly (is a 4-gram) [51].

Chapter Three

The Proposed Pattern Discovery System

Chapter Three

The proposed Pattern Discovery system

3.1 Introduction

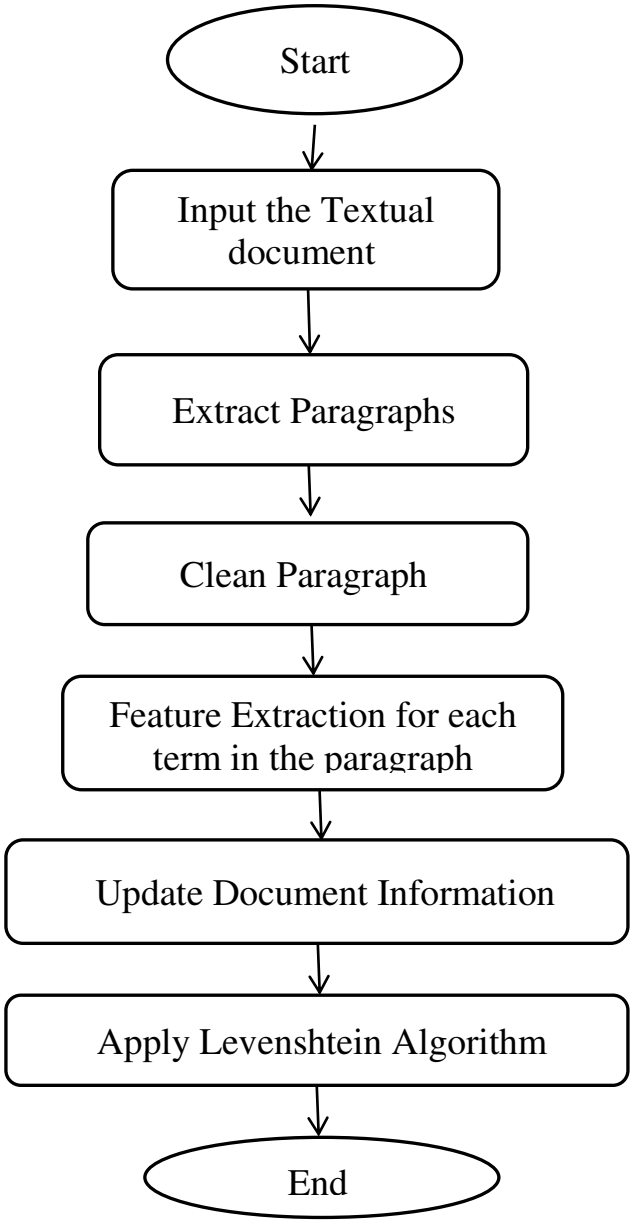
The rapid growing of digital data in the form of text document has faced some problems, analyzing these data manually is a hard task. The documents include several important terms which pointing on the valuable information additionally is supporting with various words inside the document. It is an auto-process to identify a fixed amount of key phrase or word which better reflect the main document content. Generally, it means the process of discovering useful patterns, structures and other valuable information from unstructured natural language texts. The keywords refer to the process of documents summarization which assists other words inside the text for inaugurating the main content. This process is a significant approach to analyzing data. This chapter is concerned with the design considerations, implementation requirements, and the steps taken from the establishment of an innovative solution for text mining by building a pattern discovery. The proposed system used various operations mainly relies on using machine learning and text mining techniques.

3.2 The Proposed System

Generally, the proposed system includes some basic stages to perform all relevant process for effective patterns extraction from a massive amount of unstructured data. The proposed system for unigram grammar will be explained in subsection (3.2.1) and for bigram grammar will be explained in subsection (3.2.2).

3.2.1 The Proposed System for Unigram Grammar:

The proposed system for unigram grammar includes six steps (input text, extract paragraphs, clean paragraph, feature extraction, update document information, apply Levenshtein algorithm). Figure (3.1) shows the proposed system flowchart of unigram grammar.



Figure(3.1): The Proposed System Flowchart of Unigram Grammar.

The description of each step at the figure (3.1) is given below:

1- **Input text step**: This is the first step in the proposed system. The input is entered via the end-user. The input can be a text document (txt.). The raw substance for text mining is the text documents which are unstructured since these documents do not include pre-defined relations among phrases or words when storing them in the computer. Figure(3.2) gives an example for these documents.

I love the new body style and the interior is a simple pleasure except for the center dash. However, there are a couple of things that kill it for me 1 terrible driver seat comfort, kills my back 2 lack luster interior design, my Acadia has much better comfort 3 the VCM drives me crazy because the constant change in cylinder use is perceptible enough to be an annoyance. Love the interior and the power and speed, but not hard to beat after what I had. Love the interior and exterior look, the V6 is sensational, and getting compliments on the steel metallic color as if it's a Lexus or BMW. The seats are decent, the interior design is excellent IMO as well as the exterior design, and thus far it has been extremely reliable. The interior quality is OK, my 1999 Accord EX had a better comfort level on the seats. The interior design was much nicer. The interior is nicely equipped and I like the XM radio but not the monthly fee. The new styling is very upscale, and the interior layout is also impressive and spacious inside. My only reservations are the ivory cloth interior's durability had to have taffeta white !

Figure (3.2): An Example of Text Document.

2- **Extract Paragraph Step**: This is the second step in the proposed system, In order to efficiently utilize the discovered pattern, this step is related to Pattern Taxonomy Model (PTM). This model works on re-evaluating the patterns measures via deploying them into a common hypothesis space depending on their correlations in the taxonomies of the pattern. This results in high specificity patterns to the subject which can become adequate and reasonable important values leads to an important development in the system efficiency.

PTM method firstly works on scanning the uploaded document and converting the entire document into a set of paragraphs which used as separate documents. After that, a process for extracting an amount of the set of terms from the obtained documents and terms form a specific pattern is done.

A set of paragraphs are shown in Table (3.1), to the specified document "d", here SP(d) = {dp_1, …, dp_6} the whole redundant words are removed. Considering that min-sup ≥ 2. The '5' frequent patterns are shown in Table (3.2) with its covering-sets.

Table (3.1): Paragraphs Set.

Paragraph	Terms
dp1	t3 t4
dp2	t1 t2 t3
dp3	t1 t2 t3 t6
dp4	t1 t2 t3 t6
dp5	t3 t2 t9 t6
dp6	t5 t4 t3 t2

Iteration 1

Terms	Sup.
t1	3
t2	5
t3	6
t4	2
~~t5~~	~~1~~
t6	3
~~t9~~	~~1~~

Terms	Sup.
t1,t2	3
t1,t3	3
~~t1,t4~~	~~0~~
t1,t6	2
t2,t3	5
~~t2,t4~~	~~1~~
t2,t6	3
~~t3,t4~~	~~1~~
t3,t6	3
~~t4,t6~~	~~0~~

Table (3.2): "5" Frequent Patterns with its Covering-Sets.

`Frequent Pattern	Covering Set
{t1, t2, t3}	{dp2,dp3,dp4}
{t1, t2, t6}	{dp3,dp4}
{t1,t2,t3,t6}	{dp3,dp4}
{t1, t3, t6}	{dp3,dp}
{t2, t3, t6}	{dp3,dp4,dp5}

Terms	Sup.
t1,t2,t3	3
t1,t2,t6	2
t1,t2,t3,t6	2
t1,t3,t6	2
t2,t3,t6	3

Iteration 2

Where :

Min-sup: It means that any frequency of a term less than 2 is canceled.

ti: is mean a term in the paragraph.

dpj: is the paragraph of a document " d".

There are some terms used for Pattern Taxonomy Model in the proposed system:

a-Term Frequency (TF): It is one of the main techniques for keyword extracting in which the word existence in the document is counting, for example, when TF of the word (Text) is equal to "8", this means that the term (Text) appeared "8" times in a document. Generally, if the TF is high, then the word is a significant one. TF can be calculated by using equation (2.3).

b- Term Supporting: Supposed a term set "X" in the "d" document, X is utilized for denoting the covering set of "X" to "d" document, that consists of the whole all "dp" paragraphs ∈ PS(d) like: X⊆dp, this means;

$X = \{dp | dp \in PS(d), X \subseteq dp\}$.

Where :

X– The covering set of a document d

dp – a paragraph of document d

PS(d) – a set of paragraphs for document d

c- Threshold: Threshold represents a boundary between the important terms and non-important terms. The threshold is used for reducing the number of discovered patterns in a big document. These discovered patterns of minimum relative support will maximize the training burden. In this research, threshold values from 1 to 10 are used in the tests of the proposed system, because the higher the value of the threshold as we get closer to the words of the document as a whole and this is not useful and confuse the information that extracted from the text. Figure (3.3) shows the proposed PTM flowchart for absolute support value comparsion.

In figure(3.3), PTM algorithm compares the value of absolute support (SUP_a) with the value of threshold, if the value of SUP_a is greater than the threshold value then the term will be added to the terms group and then PTM computes the accuracy.

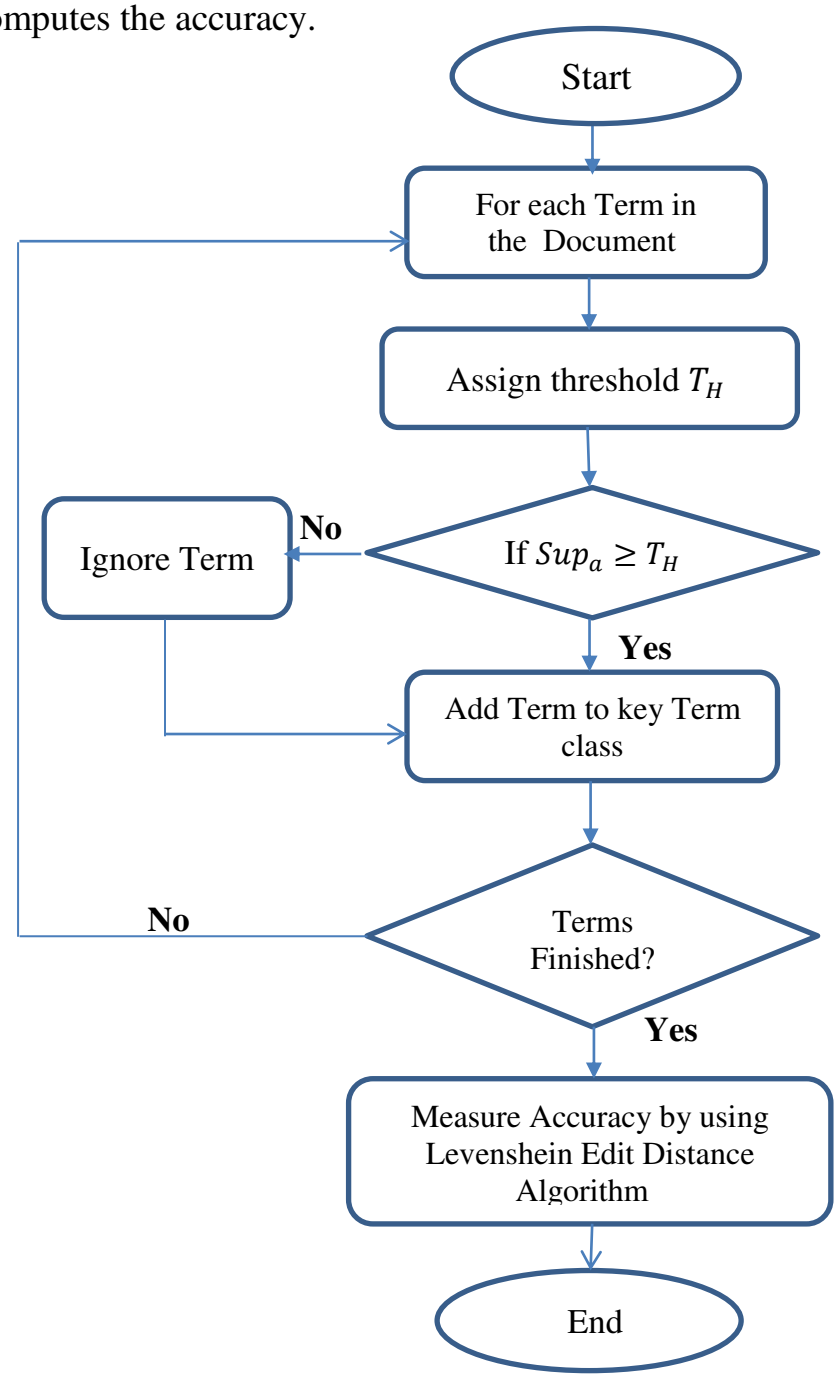


Figure (3.3): The Proposed PTM Flowchart for Absolute Support Comparsion

PTM repeats the process for the value of relative support (SUP_r) and global probability and compute the accuracy of each feature. Figure (3.4) shows the proposed PTM flowchart for relative support comparison.

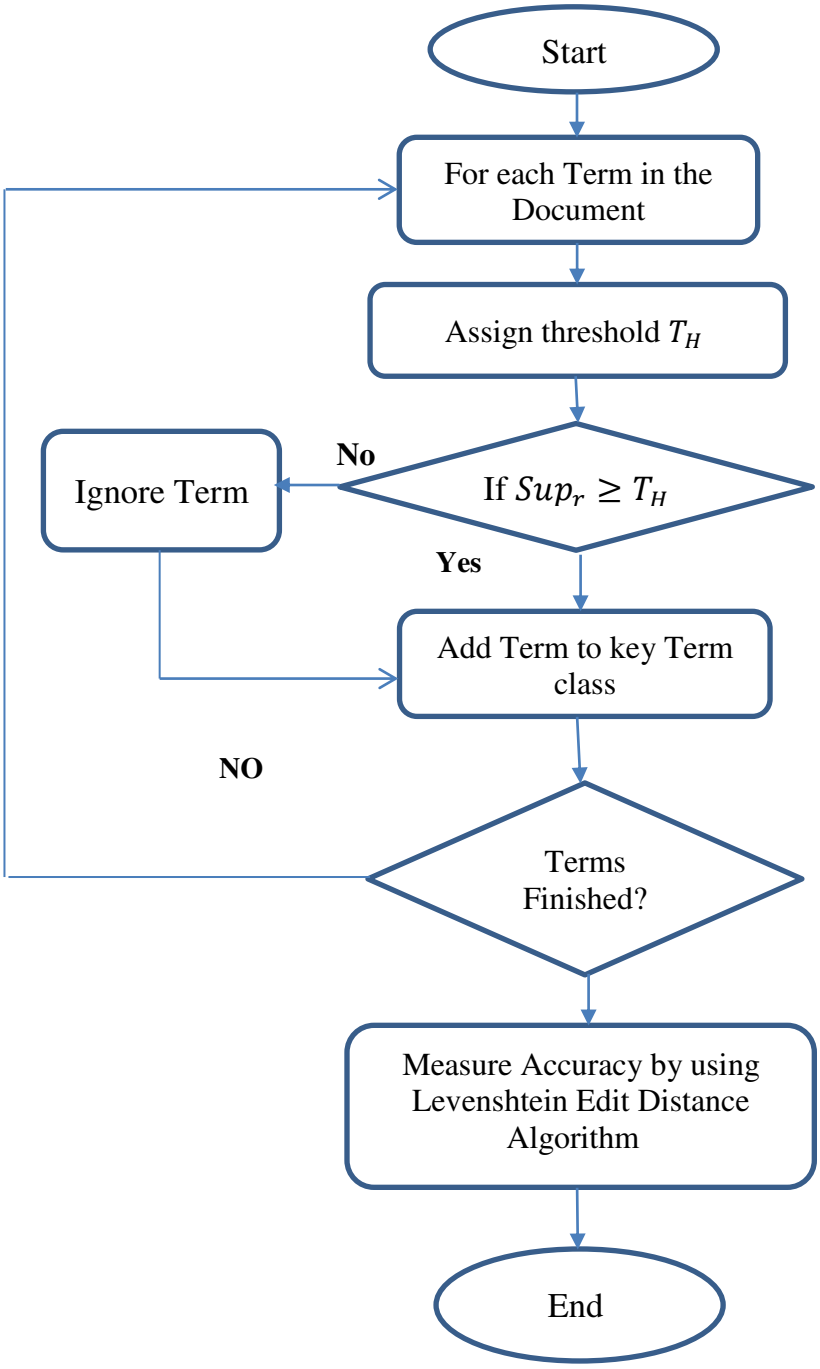


Figure (3.4): The Proposed PTM Flowchart for Relative Support Comparison

In figure(3.4), PTM algorithm compares the value of relative support (SUP_r) with the value of threshold, if the value of SUP_r is greater than the threshold value then the term will be added to the terms group and then PTM compute the accuracy. The proposed PTM model that compares the threshold value with global probability value is doing by applying Algorithm (3.1).

Algorithm (3.1)**:** The Proposed PTM Algorithm for Global Probability Comparision

Input: a set of documents, a threshold
Output: a set of paragraphs
Begin
Step1: For every term T into "d" document
Step2: Assign threshold T_H
Step3: If global probability $\geq T_H$ Then
Step4: Key Terms ←T
Step5: else Ignore T
Step6: End If
Step7: End For
Step8: Measure Accuracy by using Levenshtein Edit Distance
End

3- Clean Paragraph Step: This is the third step in the proposed system. The purpose of this step is finding the key terms in the text document. Each individual word is considered as term. Clean paragraph consist of two steps. These steps include:

i. **Removing Stop-Word:** Terms extracted from a document are usually

compared to a list of common words 'noise' words that would be useless for running queries and any matches against this list are discarded. This is usually called a stop list (the words on this list are called stop-words). Articles, prepositions, and pronouns are the most popular used words in the text documents that provide no meaning and can be considered as stop words. These words are not needful in applications of text mining, therefore, these words will be eliminated. The number of stop-words are 970 word. An example of these stop- words are "a", "an", "the", "in", "and ", " but ", " near", "to", "it", "as", "able", "about", "above", "of", "allow", "allows", "alone", "am", "an", "and", "but", "clearly", "can", "consider", and ….etc.

ii. **Applying Porter Stemming**: Algorithm(2.1) shows porter algorithm. In this process the derived words are cut down to their stem or base root words. For example learning is converted into learn, filtering is cut down to filter. Removal of "ing","ed","es" etc. comes under stemming process. Porter algorithm is used to remove the stemming from the words and convert them into their base stem. Mostly, it a written word form. In order to get the root of the word, the porter stemming can be used.

The first step of the algorithm(2.1) handles plurals, past participles, present participles. The second step transforms a terminal "y" to an "i", For example: *"generalizations"* is converted to *"generalization", "agreed"* to *"agree"*. The third step deals with double suffixes to single ones. For example: *"generalization"* is converted to *"generalize", "oscillator*" to *"oscillate"*. The fourth step removes other double suffixes not handled in the

previous step such as: *"generalize"* is changed into *"general"*. The fifth step removes remaining suffixes such as: *"general"* is transformed into *"gener"*,*"oscillate"* to *"oscill"*. The sixth step treat stems ending with –e, and treats words ending in double consonant. For example: "*attribute*" is recoded "*attribut*", *"oscill"* is converted to *"oscil"*. At the end of the sixth stage, the consequent stem is returned.

4- Feature Extraction: This is the fifth step in the proposed system. Feature extraction is an important preprocessing step, the main purpose of this step is to extract some important features from the text that was scanned by PTM. Extracted features are used to represent texts**.** Features are**:**

a- Absolute Support (Sup_a): Considering "d" document is equal to { X_1, X_2,….., X_n}, and X_i represents a sequence (a paragraph in a document). Consequently, |X| represents the number of paragraphs in "d". sup_a represents the amount of occurrence of "X" terms in SP(d). Sup_a can be calculated by using equation(2.4).

b- Relative Support (Sup_r): It is the paragraphs fraction which includes the pattern, this means the following:

"X" term set is named frequent pattern when $Sup_a(X)$ or $sup_r(X) \geq$ threshold.

Equation(2.5) can be used to compute relative support.

c- Global probability: Global probability is the probability of the term existence in the document. Global probability of a term (P) can be computed by using the equation (2.6)

Local probability: Local probability is the probability of the term existence in the paragraph. the local probability can be computed by using the equation (2.7).

e- Covering Set: is the total number of paragraphs in which term appeared.

5- Update Document information

This is the sixth step in the proposed system. For using the semantic information in the pattern taxonomy for improving the discovered pattern performance in text mining, the discovered patterns must be interpreted by summarizing them for accurately evaluating the term threshold. This process will be done by using the deploy pattern algorithm. So, a term with a higher value of TF would be no meaning when it has not cited via some significant parts of documents. Pattern deploying method has also been proposed in order to refine the patterns that help in improving the effectiveness of pattern discovery. After pattern deploying step the resulting terms is sorting according to the covering by using Timsort algorithm.

Timsort is a hybrid stable sorting algorithm, derived from merge and insertion sorts, constructed for doing a well-performing on many real-world data types. This algorithm obtains the data subsequences which are formerly ordered and utilizes that knowledge for sorting the residue more effectively. This is accomplished via merge an identified subsequence, named a run, with existing runs till certain criteria are done. Algorithm (3.2) describes the applied Deploy Pattern Algorithm.

Algorithm (3.2): Deploy Pattern Algorithm

Input: a set of terms, threshold
Output: the most frequent terms
Begin
Step1: For Terms in Document
Step2: Sort Terms using Timsort Algorithm in descending order**.**
Step3: For each Term T in Document
Step4: If GlobProb (T) > 0.0095 Then Add T to Pattern List
Step6: End if
Step7: End for
Step8: End for
End

Where

T:is a term in the document

Globprob: the global probability

From algorithm(3.2), For the purpose of obtaining more than one percent of the total words in the file, we concluded that 0.0095 is most suitable for the data studied. global probability begin from this value and Deploy pattern algorithm always test the value of global probability if it is greater than 0.0095, then the term will be added to the list of patterns.

6- Applying the Levenshtein Distance Algorithm:

This is the last step in the proposed system, algorithm (2.2) is applied for the resulting terms that are obtained from the previous step. LDA take the resulting terms with the title of the document and measure the similarity among them according to the equation (2.2).

LDA used in this research to test the efficiency of PTM algorithm and for getting more accurate results for pattern discovery. Table (3.3) is an example of the measured similarity between two short texts by using Levenshtein edit distance

Table (3.3): An Examples of the Levenshtein Distance and the Measure of Similarity Between Two Short Texts.

NO.	Text1	Text2	K	P
1	Boy	Boys	1	0.75
2	Baby	Babies	3	0.5
3	Tom is drawing a tree	Tom is draw trees	3	0.40

Where

K : is the number of the difference of characters between two words or sentences**.**

P : is the probability of similarity between two words or sentences

From table (3.3), LDA calculates the probability through measuring the similarity between two words or two sentences by calculating the number of letters of the word to the longest word between them.

3.2.2 The Proposed System for Bigram grammar:

The proposed system includes seven steps (input text, extract paragraphs, clean paragraph, applying bigram grammar, feature extraction, update document information, and Apply Levenshtein algorithm). Figure (3.5) shows the flowchart of the proposed system for Bigram grammar.

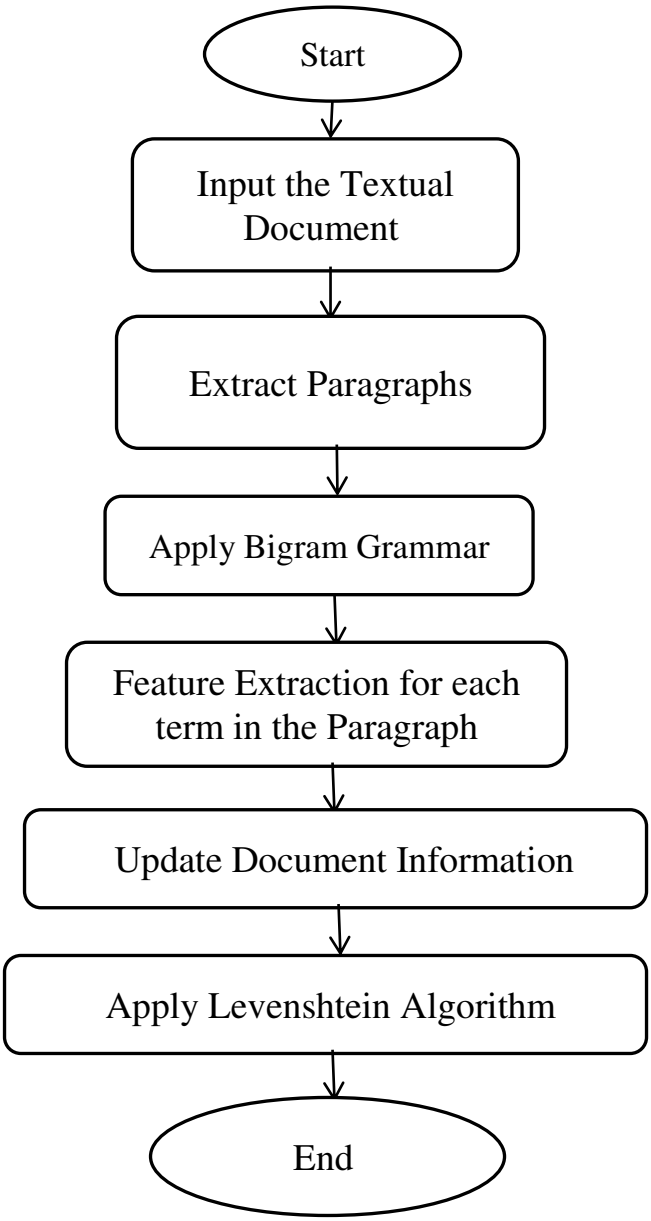


Figure (3.5): The Proposed System Flowchart for Bigram Grammar.

The steps of bigram grammar for the proposed system are the same as unigram grammar except:

4- Apply bigram grammar: After applying stop–word removal and porter stemmer, in this step the document is analyzed so that it takes two words each time. It takes the first and the second term and then the second term and the third term and etc., The result of analyzing of the document is two words in every time. Figure (3.6) is an example of applying the bigram grammar step for the text.

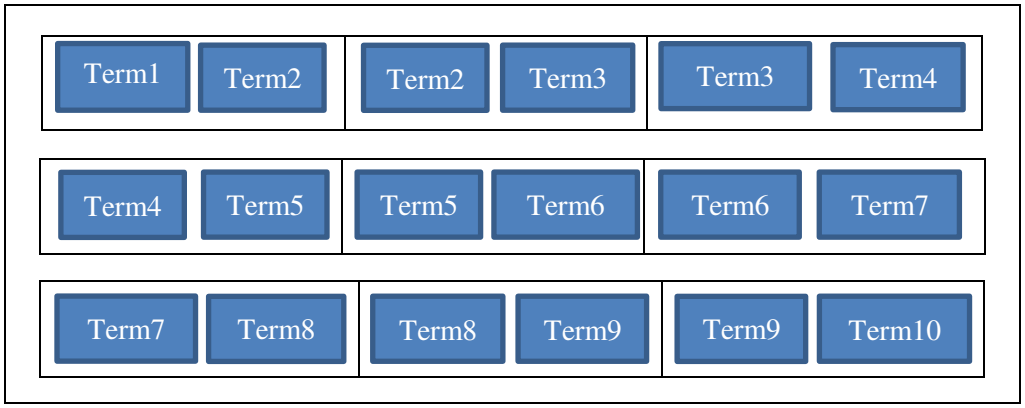


Figure (3.6): An Example of Applying Bigram Grammar for the Text.

3.3 Accuracy

The equation to compute the accuracy of each document (file) is as the following:

$$\text{Accuracy}=1-\left(\frac{\mathbf{P}}{\mathbf{size\ of\ detected\ terms}}\right)*100\% \qquad (3.1)$$

Where

P : is the probability of Levenshtein edit distance algorithm.

Size of detected terms : the number of detected terms.

3.4 Average of Accuracy

The average accuracy of a dataset can be computed by using the following equation:

$$\text{Average Accuracy} = \frac{\textbf{Total Accuracy}}{\textbf{Counter}} * 100\% \quad (3.2)$$

Where :

Total Accuracy: the total accuracy for each document.

Counter: the number of documents.

3.5 The Total Time of Processing

For accurate processing of a text, we compute the time of processing of a document in each step in the proposed system by using equation (3.3):

$$\text{Time of Processing} = \text{End time} - \text{Start time} \quad (3.3)$$

Where:

End time: is the end time for processing the document.

Start time: is the start time to process the document.

After finding the time of processing for each document, we calculate the total time to extract patterns from a data set by using equation (3.4):

$$\text{Total Time} = \text{The sum of times of processing for each document} \quad (3.4)$$

Chapter Four

The Experiential Results and Tests

Chapter Four

The Experiential Results and Tests

4.1 Introduction

This chapter summarizes the implementation outcomes obtained by the developed system described in detail in Chapter three. The experimental results and tests of the system phases will be explained. In other words, this chapter is the evaluation of the performance of the proposed pattern discovery system. It will contain a detailed depiction of the steps involved in application implementation.

4.2 The Environment of Implementation

The implementation of the developed system is done in java 13.0.1 and Eclipse 2018-12 programming language using a laptop computer with windows7 ultimate. The experiments were performed on an Intel (R) Core (TM) i5-4210U CPU @1.70 GHz 2.40 GHz, 64-bit Operating System, and 8GB RAM. In the following sections, the detailed steps and implementation results will be explained for each step to accomplish the suggested system.

4.3 Datasets

The implementation and testing are applying on two datasets:

1- Opinosis Opinion Dataset 1.0 – Documentation (dataset1): The dataset1 includes the extract sentences from surveys in a specific topic. As an instance for topics; *"room holiday in London"* and *"navigation amazon kindle"*. Each topic or document is stored in the form of a .txt file and has a specific title. There are 51 such topics in this dataset and for each topic, there are about one hundred sentences. The opinions are obtained from various sources such as

Trip advisor(hotels), Amazon.com(various electronics), and Edmunds.com (cars). The total size of dataset1 is 28672 Bytes with different lengths of text.

2- **Reuter_50_50 Dataset** (dataset2): The dataset2 is a subset of RCV1. These corpus has already been used in author identification experiments. In the top 50 authors (with respect to the total size of articles) were selected. 50 authors of texts labeled with at least one subtopic of the class Criteria Cognitive Aptitude Test (CCAT) (corporate/industrial) were selected. That way, it is attempted to minimize the topic factor in distinguishing among the texts. The training corpus consists of 2,500 texts (50 text per author) and the test corpus includes other 2,500 texts (50 per author) non-overlapping with the training texts**.** The overall size of dataset2 is 1048576 Bytes.

4.4 Proposed System Implementation for Unigram grammar

To calculate the accuracy of a document**,** the system has seven stages executed sequentially, starts by input text file and ends with Apply Levenshtein algorithm as described in the following sections**.**

4.4.1 Input Text

The system gets the text from a dataset which is stored on the computer. Usually, load text from a folder "Openosis or Routers 50_50 dataset", this folder is stored in a file of ".txt" or text document to increase access speed during system execution**.** Figure (4.1) illustrates the input text step in the proposed system.

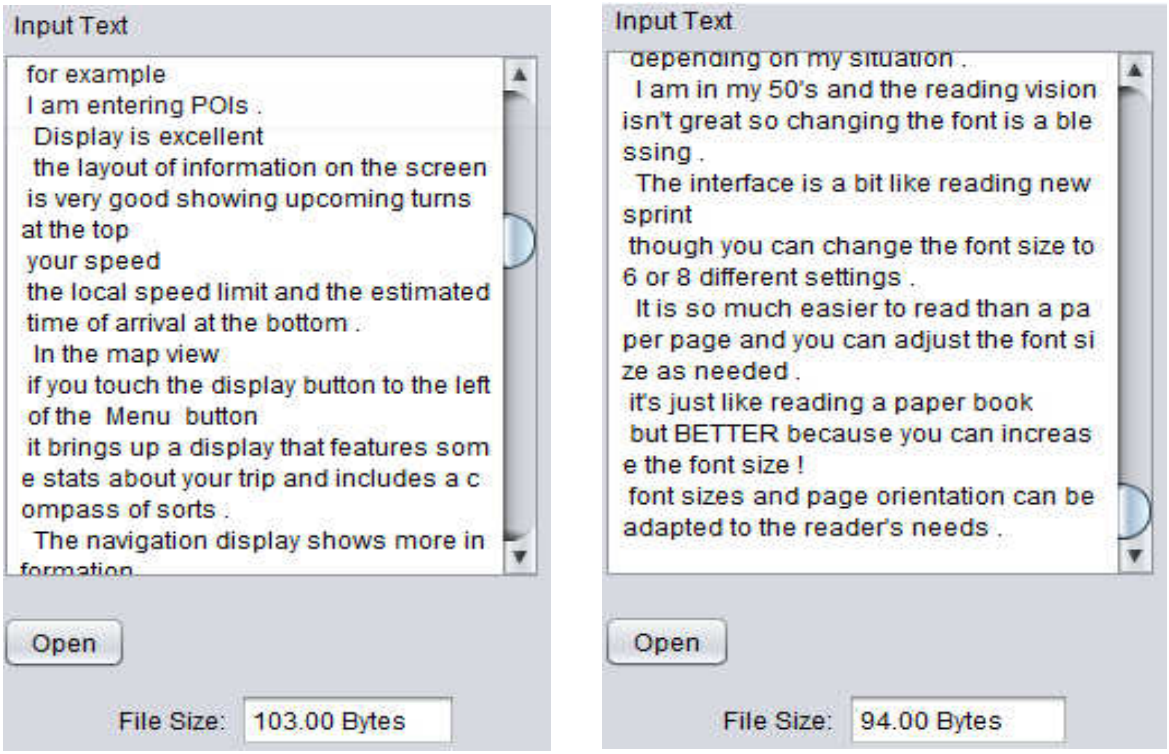
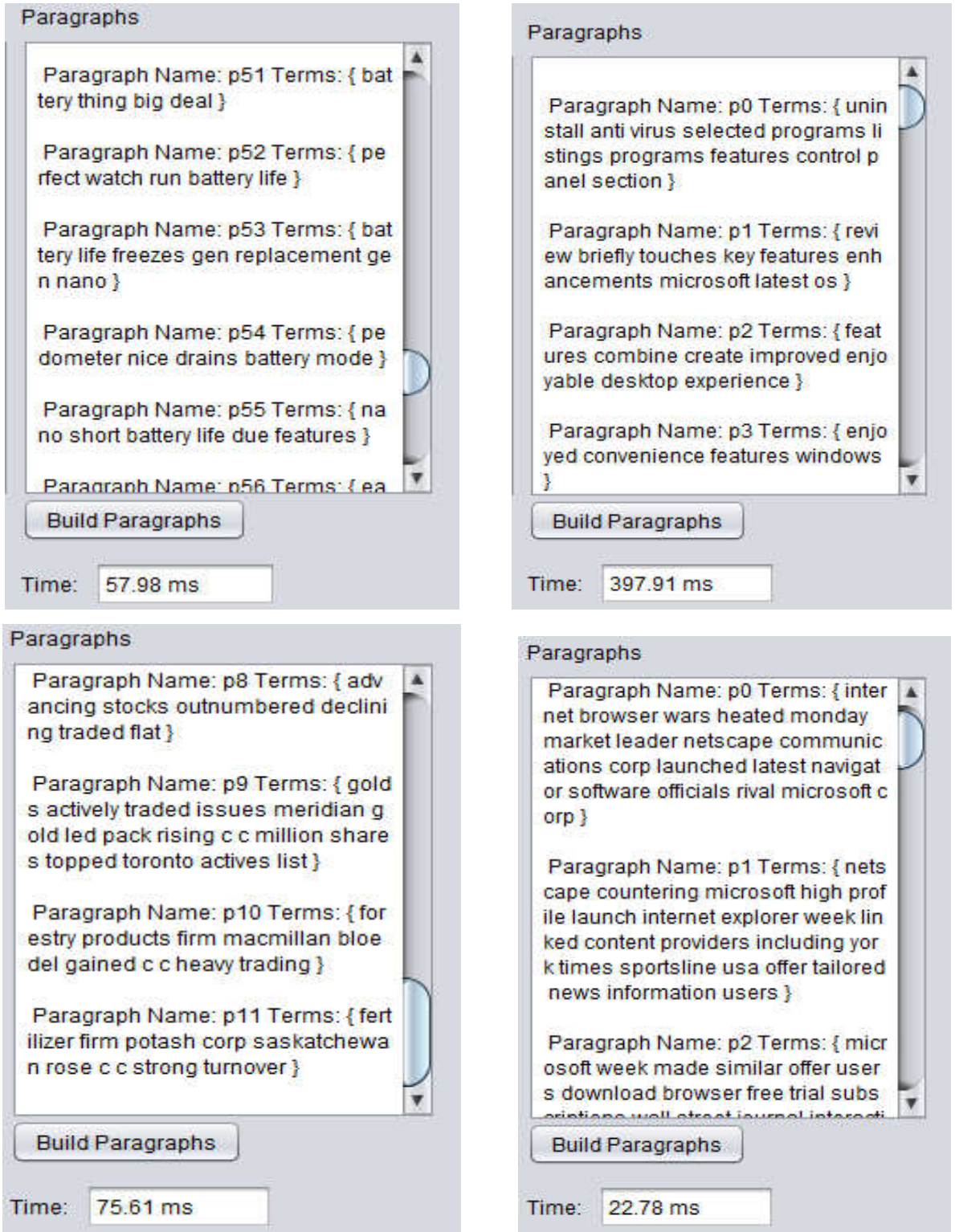


Figure (4.1): An Example of an Input Text Step

In fig(4.1), the proposed system load the text document and read it to perform the next step.

4.4.2 Extract paragraph Step

In this step, after loading the text, PTM algorithm is performed on the text which implemented in the flowchart in figure (3.3), figure (3.4) and algorithm (3.1). In this step the value of threshold between the range[1-10] to get from 1% to 10% the information in the text. The value of threshold will be compared with the value of global probability, absolute support, and the relative support for the two datasets. After the comparison, stop-words will be eliminated and porter stemming will apply. The result of this step is that each word in each paragraph is considered as a term. Figure(4.2) states the division of the text into paragraphs.

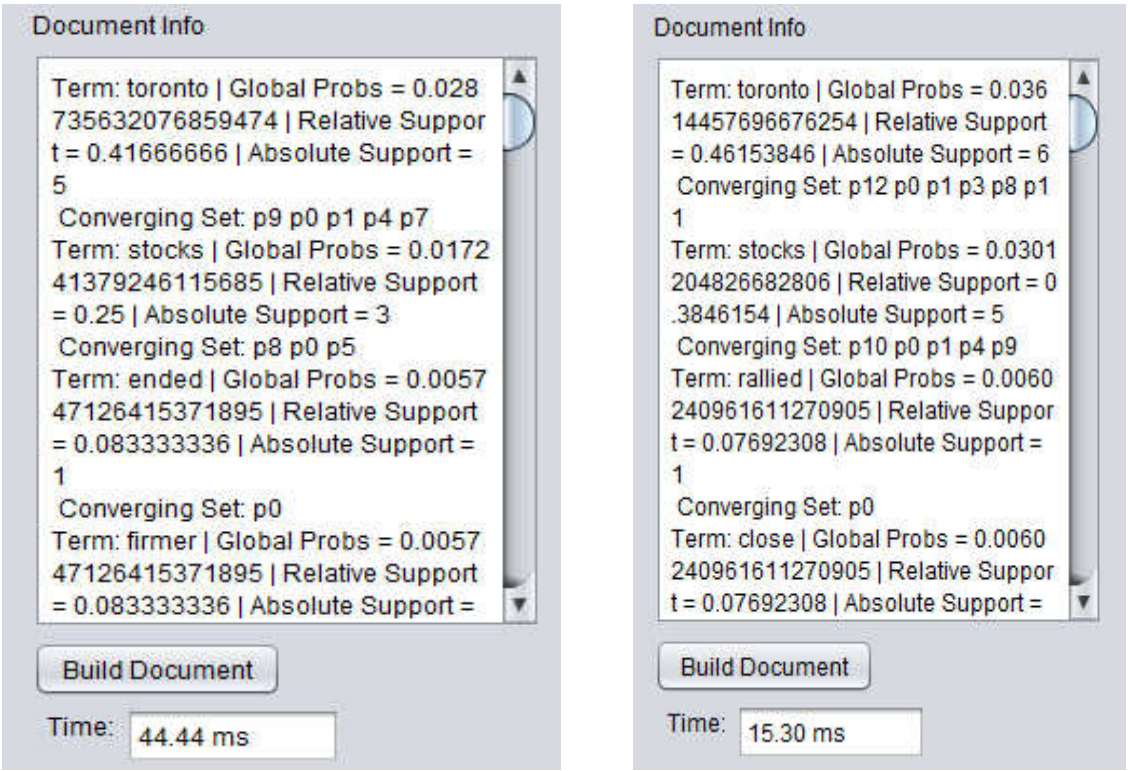


Figure(4.2): The Division of Text into Paragraphs

From fig(4.2), the document is divided into number of paragraphs and each paragraph treated as an individual document which consist of set of terms.

4.4.3 Feature Extraction step

In this step, five types of features were calculated for each term by implementing equation (2.4),(2.5),(2.6), (2.7)and (2.8). These features are a global probability, local probability, absolute support, covering set and relative support. Figure (4.3) illustrate the calculation of the five features for each term in the document.

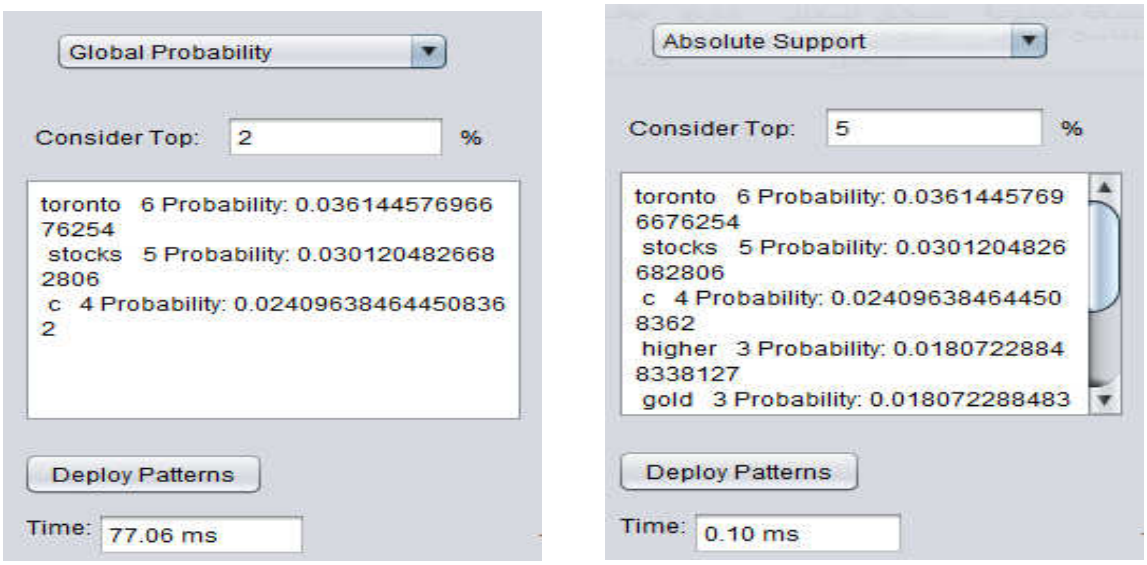


Figure(4.3): The Calculation Process of Features for Each Term in the Document.

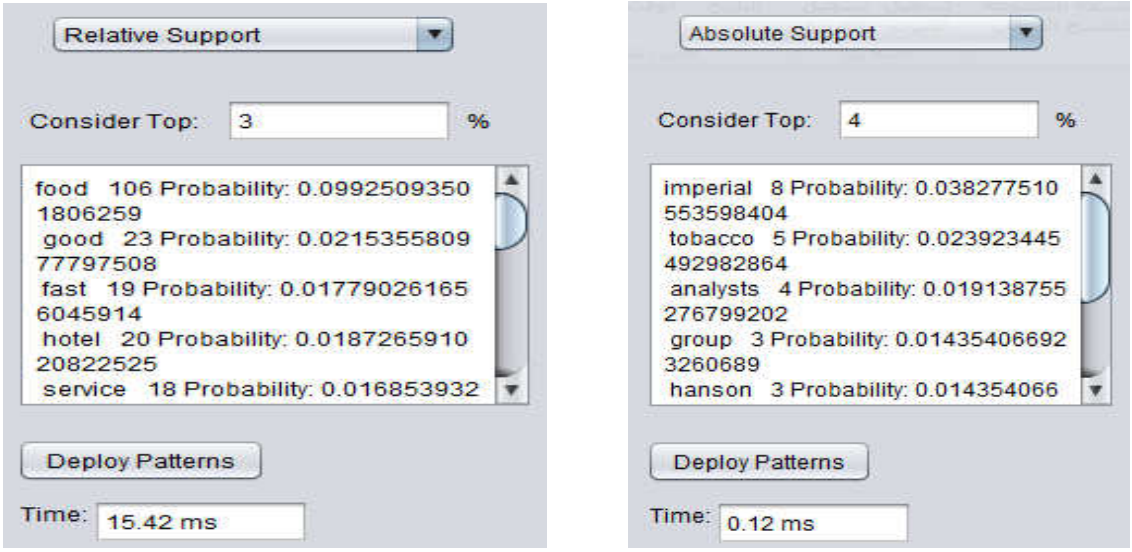
4.4.4 Update Document Information(Deploy Pattern)

After calculating the five features of a term, by applying algorithm (3.2), the most frequency term is considered as the word that has the full meaning of a document and important. In this step, the terms with high frequency based on

the threshold value for one of the four features (global probability, absolute support, local probability and relative support) are arranged in descending order. Figures (4.4) and (4.5) is an example of the most frequent terms of a document for various threshold values for dataset1 and dataset2 respectively.



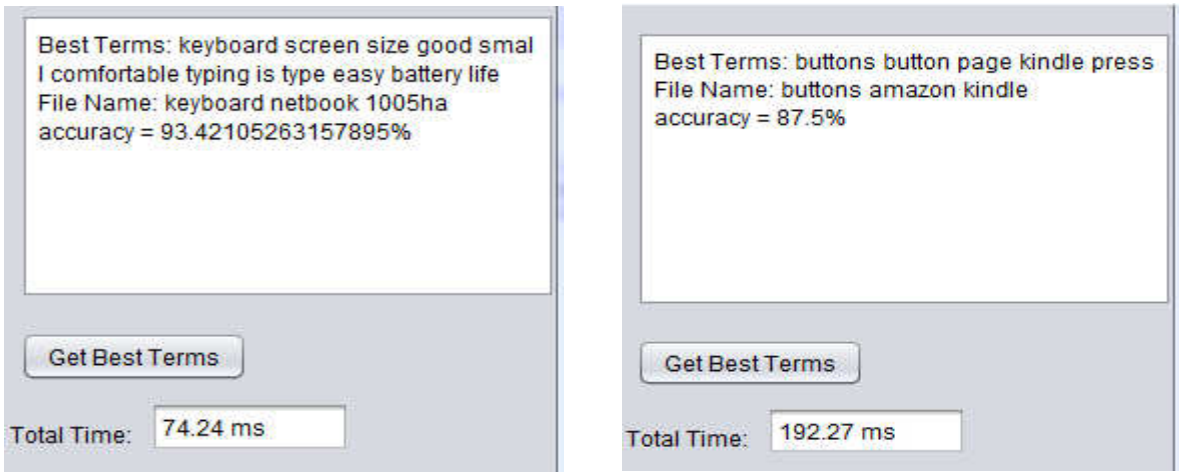
Figure(4.4): An Example of the Most Frequent Terms of a Document for Various Threshold Values for Dataset1



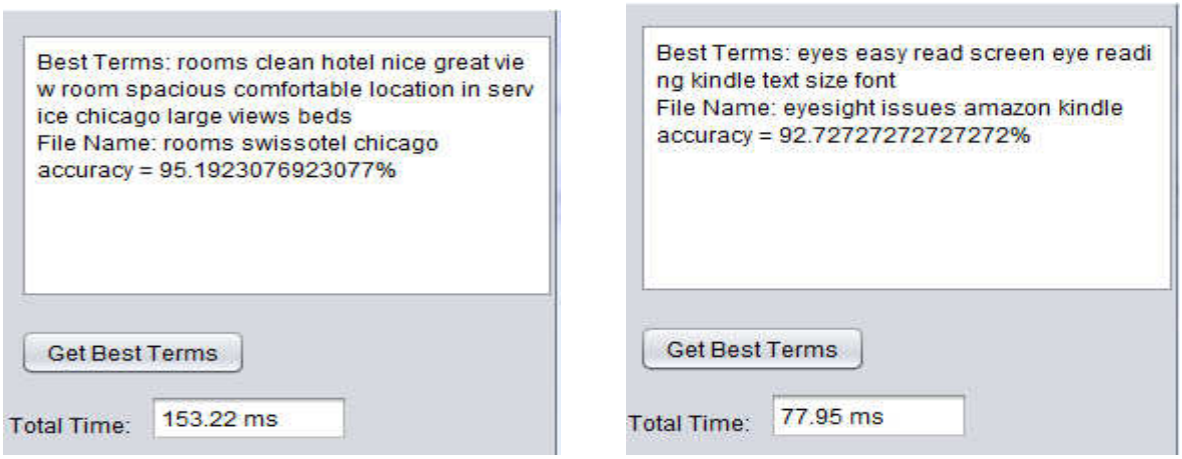
Figure(4.5): An Example of the Most Frequent Terms of a Document for Various Threshold Values for Dataset2

4.4.5 Applying Levenshtein Distance algorithm

By applying the similarity equation which is described previously in equation (2.2) on the resulting terms from the previous step, we can get the accuracy of each document by implement equation (3.1). Figure(4.6) state calculating the accuracy of a document when the threshold is 1 for the global probability of dataset1. Figure (4.7) states calculating the accuracy of a document when the threshold is 3 for absolute support of dataset2.



Figure(4.6): The Calculation of Accuracy for Different Documents when the Threshold is 1 for Global Probability for Dataset1.

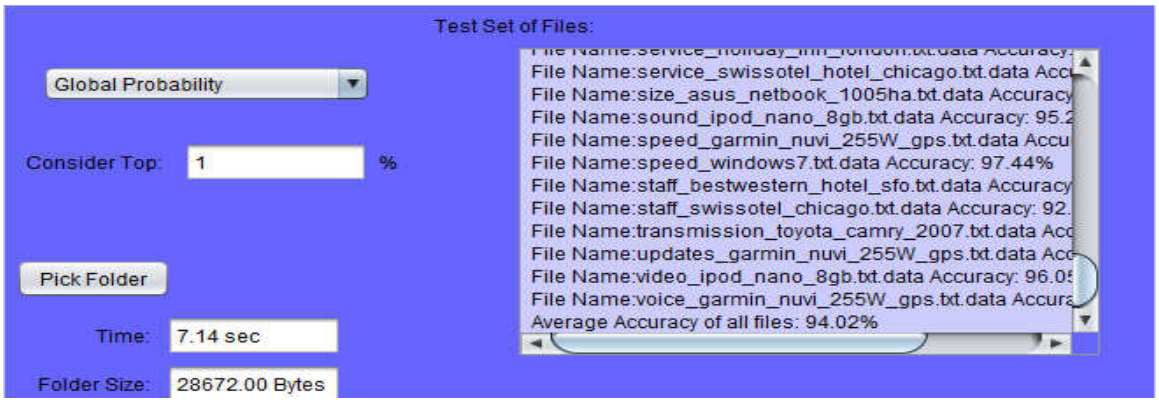


Figure(4.7): The Calculation of Accuracy for Different Documents when the Threshold is 3 for Absolute Support for Dataset2.

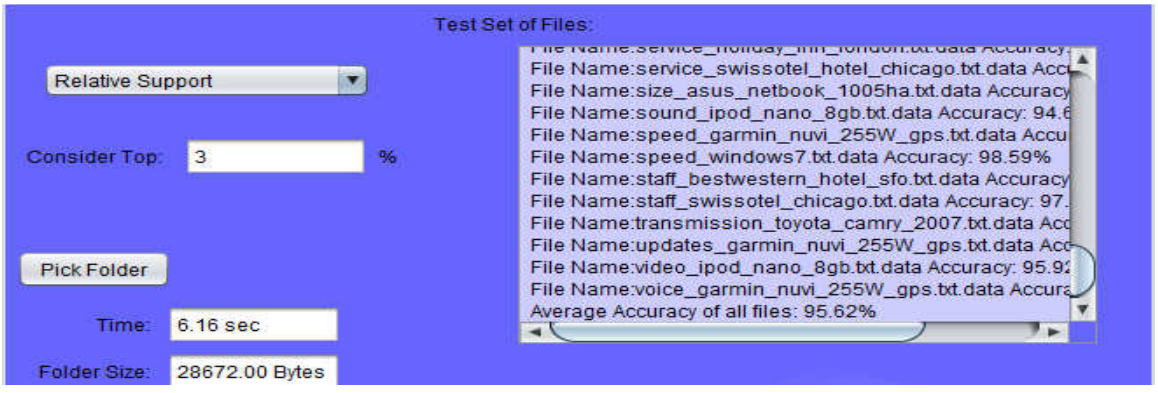
In fig(4.6) and (4.7), levenshtein edit distance is implementing on the resulting terms and the title of a document to acquire the accuracy of a document.

4.5 The Average Accuracy and Time of a Dataset

In this step, after calculating the accuracy of each document in the dataset, equation(3.2) is implemented to compute the average accuracy of a dataset. Figures (4.8) and (4.9) are an example of calculating the average accuracy of a dataset1 and dataset2 for different threshold values.

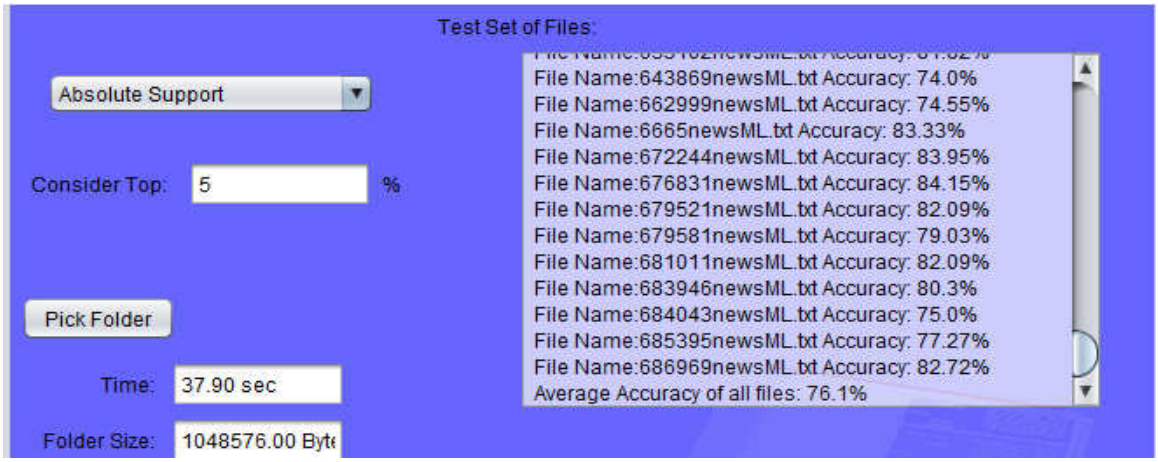


(a)

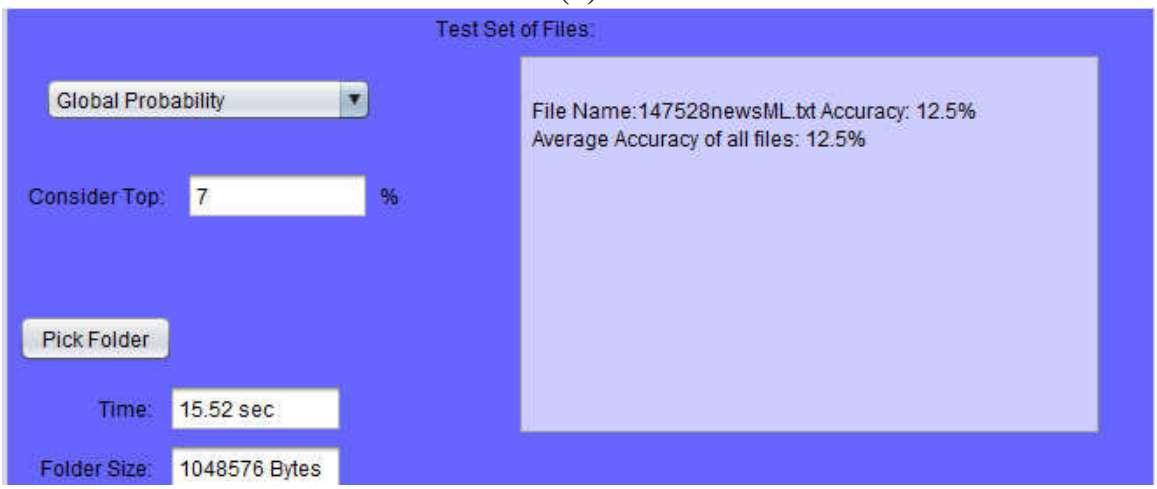


(b)

Figure (4.8): An Example of Calculating the Average Accuracy of a Dataset1 for Different Threshold Values,(a) threshold=(1) for global probability,(b) threshold=(3) for relative support.



(a)



(b)

Figure (4.9): An Example of Calculating the Average Accuracy of a Dataset2 for Different Threshold Values,(a) threshold=(5) for absolute support,(b) threshold=(7) for global probability.

4.5.1 Results by Average Accuracy and Time for Dataset1

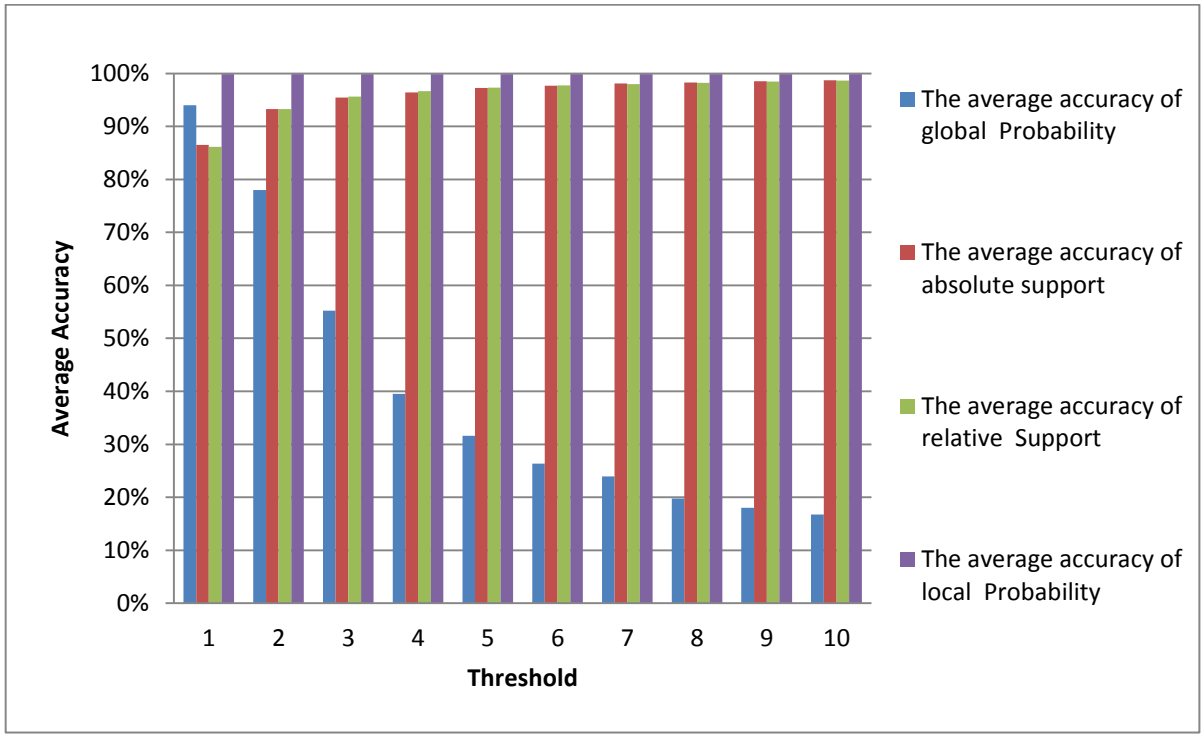
This subsection shows the results of finding the average accuracy of the proposed system for dataset1 based on the value of the threshold for one of four features because from the experiments, the proposed system got that the relative support is equal to covering set for each text. Table (4.1) describe in details the average accuracy and time of processing for the dataset1.

Table(4.1): The Average Accuracy and Time of the Proposed System on the Dataset1 for Unigram Grammar.

Threshold	The average accuracy of global Probability	Elapse Time In sec	The average accuracy of absolute support	Elapse Time In sec	The average accuracy of relative Support	Elapse Time In sec	The average accuracy of local Probability	Elapse Time In sec
1	94.02%	5.84	86.48%	6.77	86.14%	5.98	99.88%	6.09
2	78.01%	5.74	93.28%	5.96	93.28%	5.89	99.88%	5.83
3	55.19%	6.20	95.45%	6.60	95.62%	5.77	99.88%	5.83
4	39.5%	4.96	96.43%	6.75	96.63%	5.21	99.88%	5.72
5	31.61%	5.33	97.26%	4.98	97.31%	4.89	99.87%	5.77
6	26.36%	4.94	97.68%	4.98	97.74%	4.89	99.87%	5.86
7	23.95%	5.04	98.12%	4.93	97.97%	4.92	99.85%	5.85
8	19.74%	4.82	98.31%	5.24	98.24%	5.30	99.85%	5.71
9	18.02%	4.81	98.51%	5.05	98.44%	4.93	99.84%	5.77
10	16.72%	4.81	98.68%	5.17	98.63%	4.89	99.83%	5.89

From table (4.1), the average accuracy of global probability for threshold values from 1 to 10 are decreasing from 94.02% to 16.72% because the probability of the term appearing in the document decreases as the threshold value increases, the average accuracy of absolute support for threshold values from 1 to 10 are increasing from 86.48% to 98.68% because the appearing of term in the paragraph increases as the threshold value increases, and the average accuracy of relative support for threshold values from 1 to 10 are increasing from 86.14% to 98.63% because the appearing of term in the fraction of paragraph increases as the threshold value increases. From the above table, we note that the values of local probability from 1 to 10 are higher than the values of the rest features, but these results inappropriate because it takes all the words of paragraphs and calculates the probability without reducing them, and this is because of the nature of this feature and this

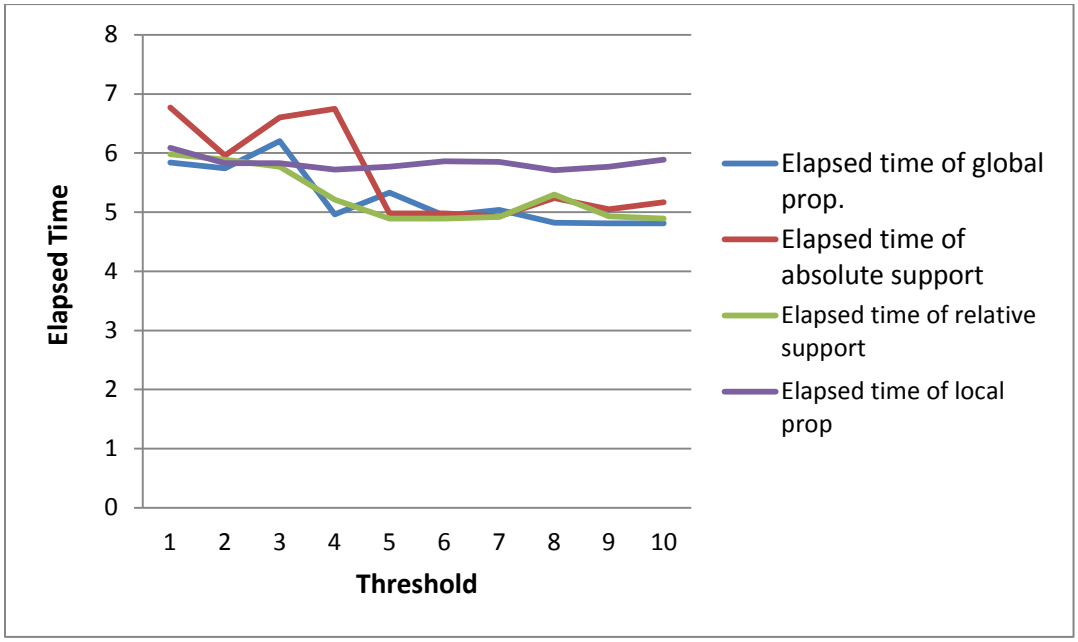
is not suitable for text mining. Table (4.1) shows that time is convergent and decreases slightly with increased threshold. Through testing on the proposed system , when the threshold value is 10 for absolute support, it got the highest average accuracy which is 98.68% for pattern discovery in text mining. Figure(4.10) states the relationship between threshold values and features values for dataset1.



Figure(4.10): The Relationship between Threshold Values and Features Values for Dataset1.

Figure (4.10) illustrate that when the value of Threshold = 10, the proposed system acquire a highest average accuracy for absolute and relative support of a dataset1. Figure (4.11) shows the relationship between the threshold value and elapsed time. Figures ((4.12), (4.13), (4.14), and (4.15)) respectively states graph representation for different threshold values.

Figures (4.15) and (4.16) prove that the value of relative support and covering set are equal for every threshold value in the proposed system.



Figure(4.11): The Relationship between Threshold Values and Elapsed Time

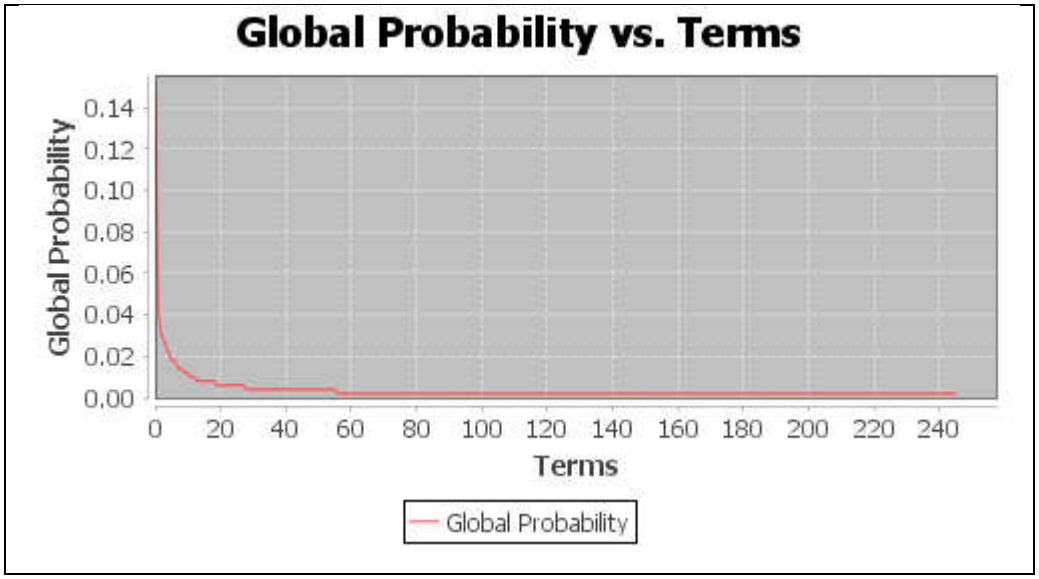


Figure (4.12): An Example of a Graph Representation of Global Probability when the Threshold Value =1.

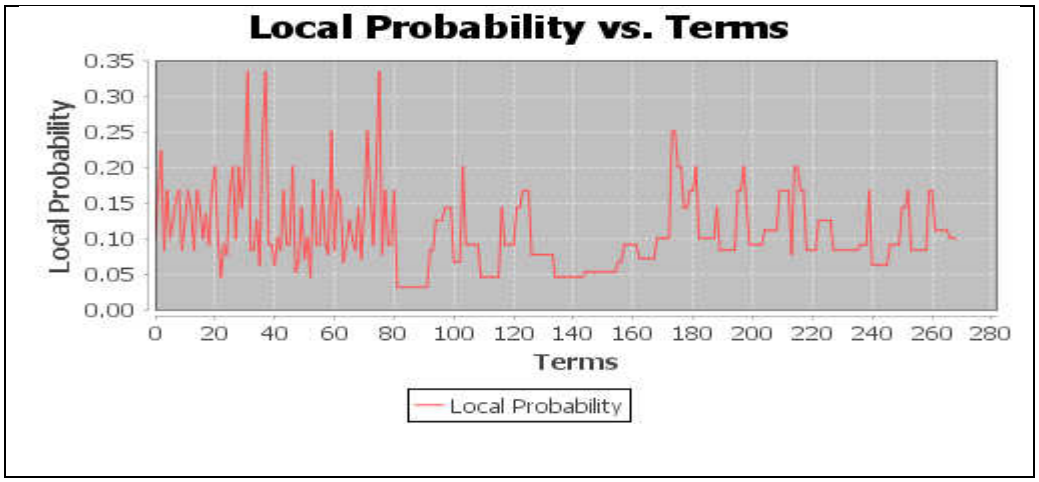


Figure (4.13): An Example of a Graph Representation of Local Probability when Threshold Value =1.

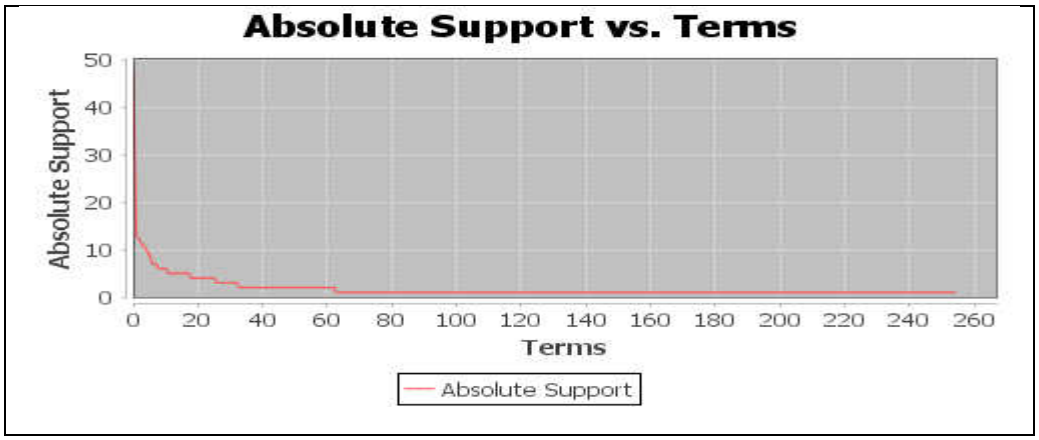


Figure (4.14): An Example of a Graph Representation of Absolute Support when Threshold Value =7.

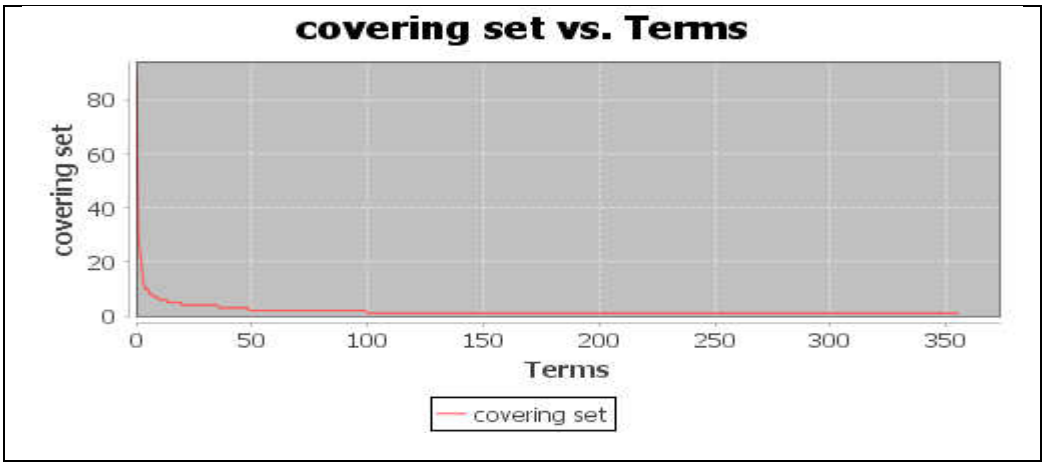


Figure (4.15): An Example of a Graph Representation of the Covering Set when Threshold Value =5.

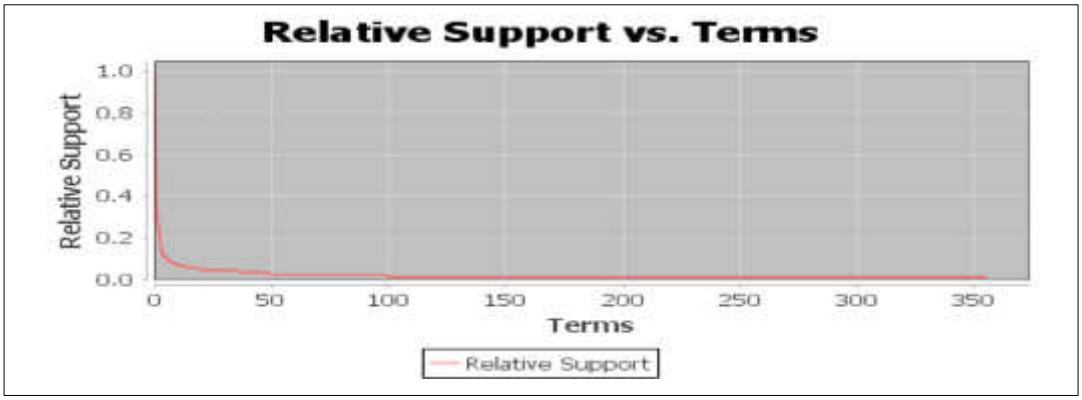


Figure (4.16): An Example of a Graph Representation of the Relative Support when Threshold Value =5.

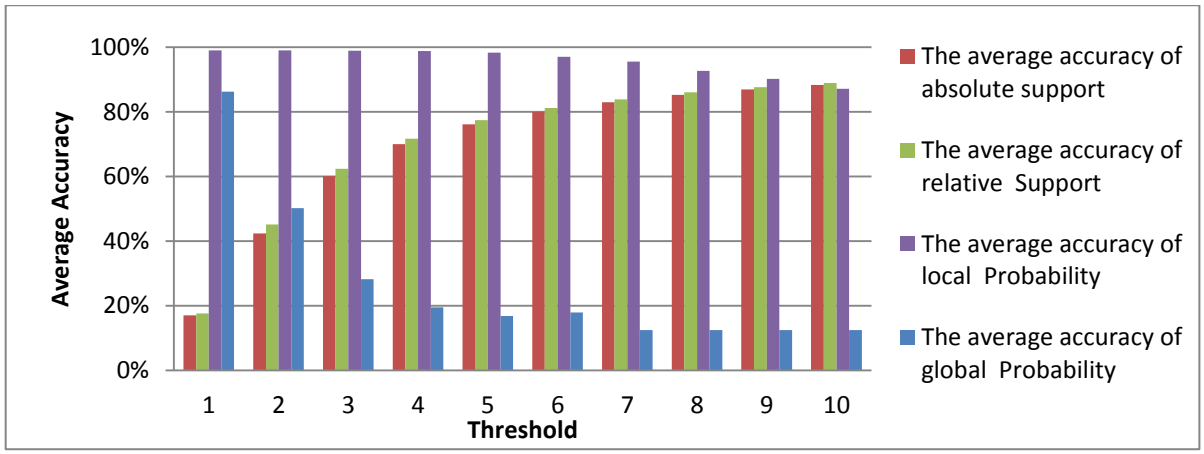
4.5.2 Results by Average Accuracy and Time for Dataset2

The results of the proposed system on dataset2 is implemented in this subsection. From the experiments, the proposed system got that the relative support is equal to covering set for each text. Table (4.2) describes in details the average accuracy and time of processing for the dataset2.

Table(4.2): The Average Accuracy and Time of the Proposed System on the Dataset2 for Unigram Grammar

Threshold	The average accuracy of global Probability	Elapse Time In sec	The average accuracy of absolute support	Elapse Time In sec	The average accuracy of relative Support	Elapse Time In sec	The average accuracy of local Probability	Elapse Time In sec
1	86.26%	30.30	16.97%	18.72	17.66%	19.52	98.98%	33.93
2	50.18%	24.87	42.35%	32.57	45.1%	29.20	98.98%	30.39
3	28.25%	18.74	60.05%	35.74	62.37	29.30	98.94%	35.35
4	19.54%	15.74	70.01%	28.64	71.7%	35.80	98.75%	37.47
5	16.83%	15.06	76.1%	30.73	77.45%	35.58	98.31%	34.63
6	17.93%	14.90	80.03%	37.73	81.17%	29.08	96.98%	34.64
7	12.5%	14.98	82.97%	36.77	83.87%	26.61	95.51%	33.70
8	12.5%	14.79	85.2%	34.23	86.01%	28.38	92.62%	29.86
9	12.5%	14.69	86.92%	33.27	87.58%	33.54	90.16%	34.27
10	12.5%	14.53	88.34%	24.52	88.89%	34.09	87.15%	30.98

From table(4.2), the average accuracy of global probability for the threshold value from 1 to 10 are decreasing from 86.26% to 12.5% because the probability of the term appearing in the document decreases as the threshold value increases. The average accuracy of absolute support for threshold values from 1 to 10 are increasing from 16.97% to 88.34% because of the appearing of a term in the paragraph increases as the threshold value increases, and the average accuracy of relative support for threshold values from 1 to 10 are increasing from 17.66% to 88.89% because the appearing of a term in the fraction of paragraph increases as the threshold value increases. From the table above, we note that the values of local probability from 1 to 10 are higher than the values of the rest features, but these results inappropriate because it takes all the words of paragraphs and calculates the probability without reducing them, and this is because of the nature of this feature and this is not appropriate for text mining. Table (4.2) shows that time is convergent and decreases slightly with an increased threshold. Through testing on the proposed system for dataset2, when the threshold value is 10 for relative support, the proposed system acquire a higher average accuracy. Figure(4.17) states the relationship of threshold values and features values for dataset2.



Figure(4.17): The Relationship between Threshold Values and Features Values for Dataset2

Figure (4.18) shows the relationship between the threshold value and Elapse time. Figures ((4.19), (4.20), (4.21), and (4.22)) respectively state graph representation for different threshold values for the dataset2. Figures (4.19) and (4.21) prove that the value of relative support and covering set are equal for every threshold value.

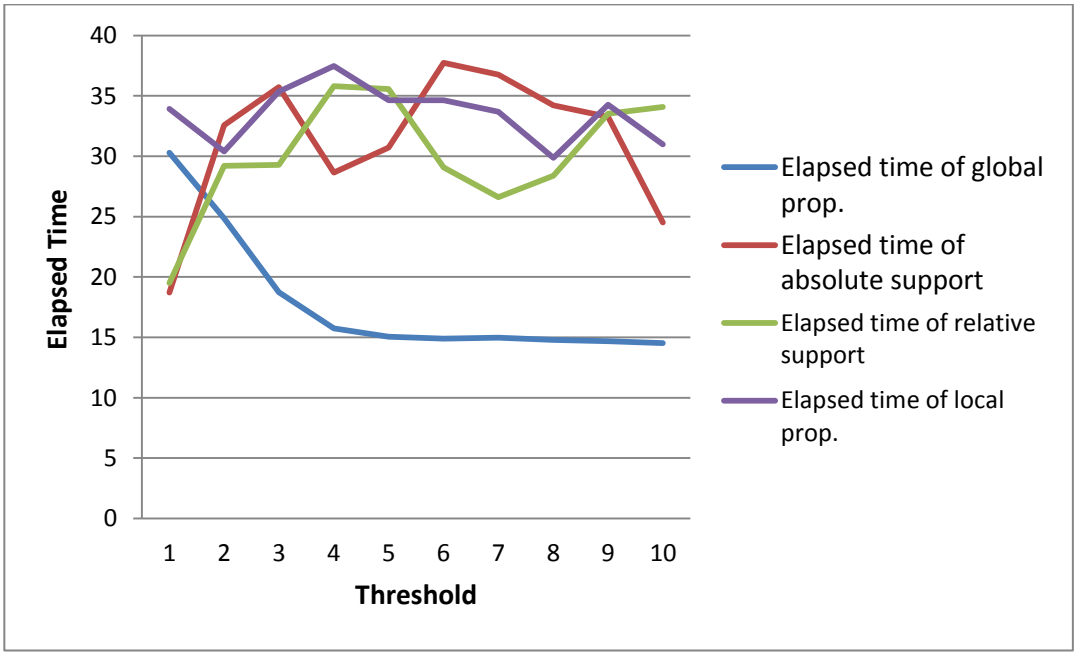
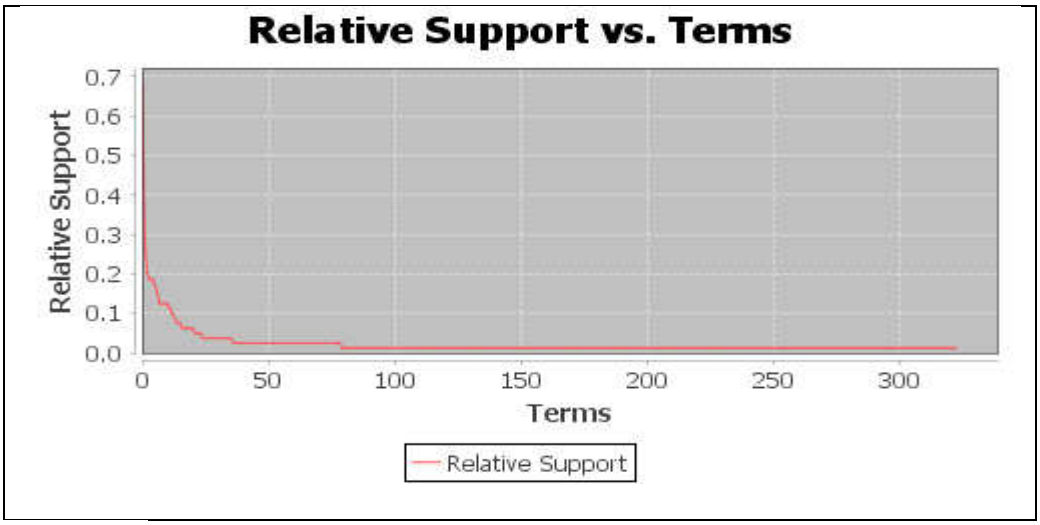
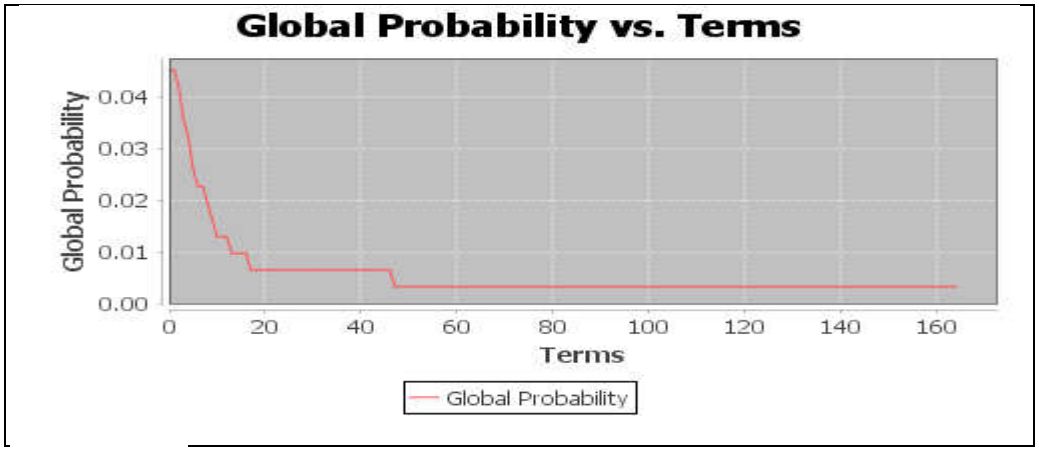


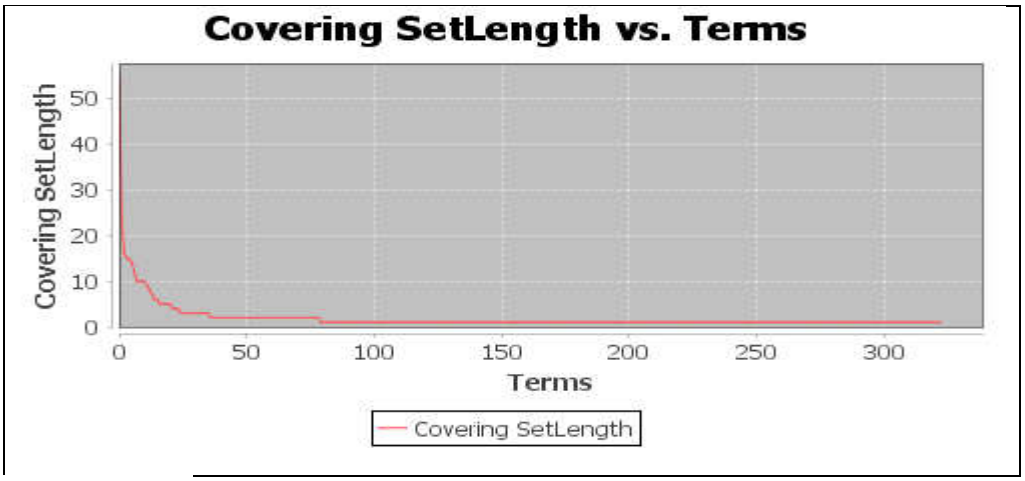
Figure (4.18): The Relationship between the Threshold Value and Elapse Time



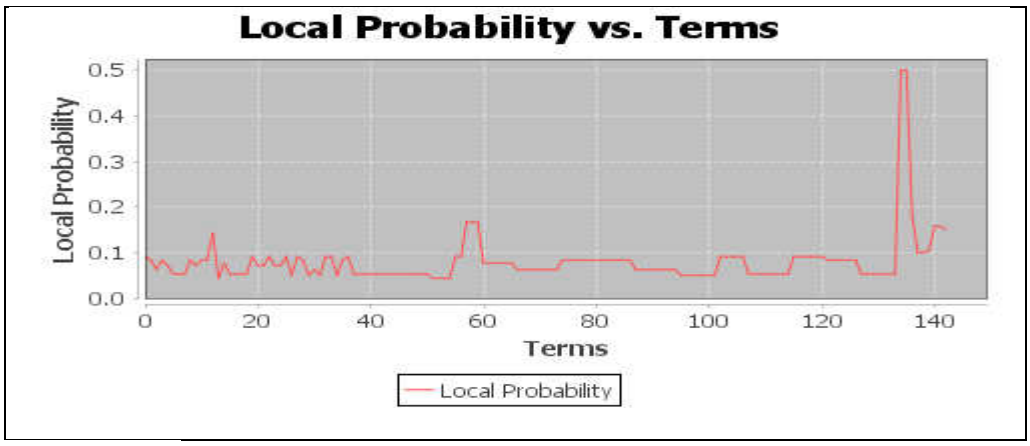
Figure(4.19): An Example of a Graph Representation of the Relative Support when Threshold Value =2.



Figure(4.20): An Example of a Graph Representation of the Global Probability when Threshold Value =8.



Figure(4.21): An Example of a Graph Representation of the Covering Set when Threshold Value =2.



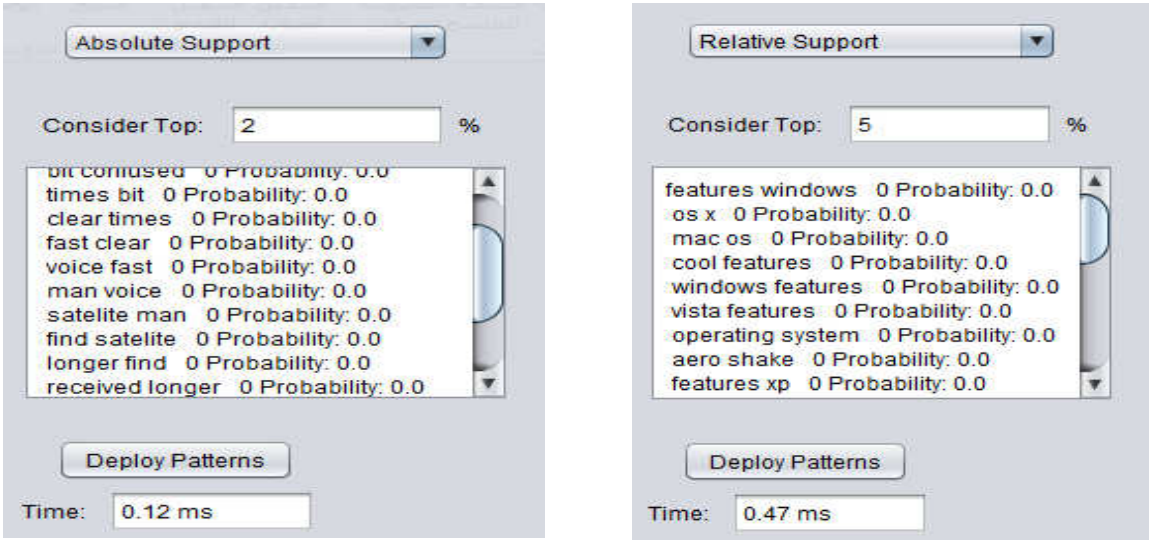
Figure(4.22): An Example of a Graph Representation of the Local Probability when Threshold Value =5.

4.6 Proposed System Implementation for Bigram grammar

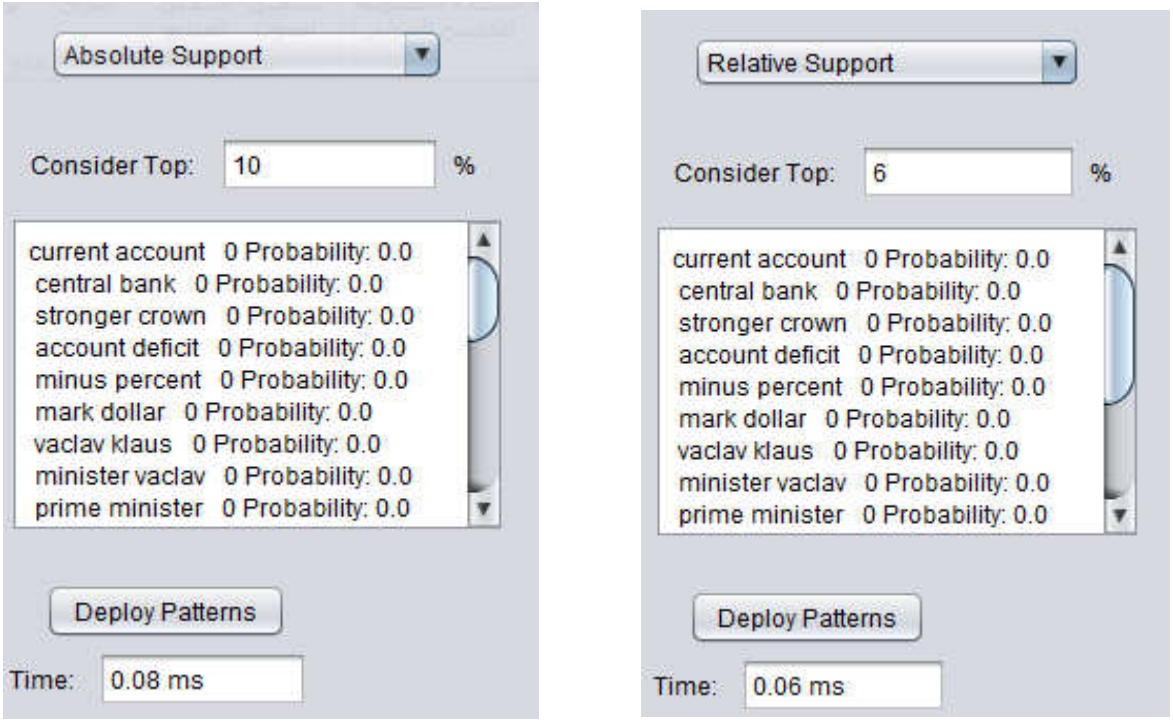
The implementation of the proposed system for bigram is similar to the proposed system for unigram except that the output of some steps. So, the input text step, extract paragraph step and feature extraction step are similar to the processing of unigram but the differences are in:

4.6.1 Update Document Information (Deploy Pattern)

In this step, After calculating the five features per term by applying algorithm (3.2), here the output of this step is two terms instead of one in the unigram grammar, the two terms that have high frequency based on the threshold value for one of the four features(global probability, absolute support, local probability and relative support) are arranged in descending order because from the experiments, the proposed system got that the relative support is equal to covering set for each text. Figures(4.23) and (4.24) are an example of the most frequent terms of a document for various threshold values for dataset1 and dataset2 sequentially.



Figure(4.23): An Example of the Most Frequent Terms of a Document for Various Threshold Values and Features Types for Dataset1.

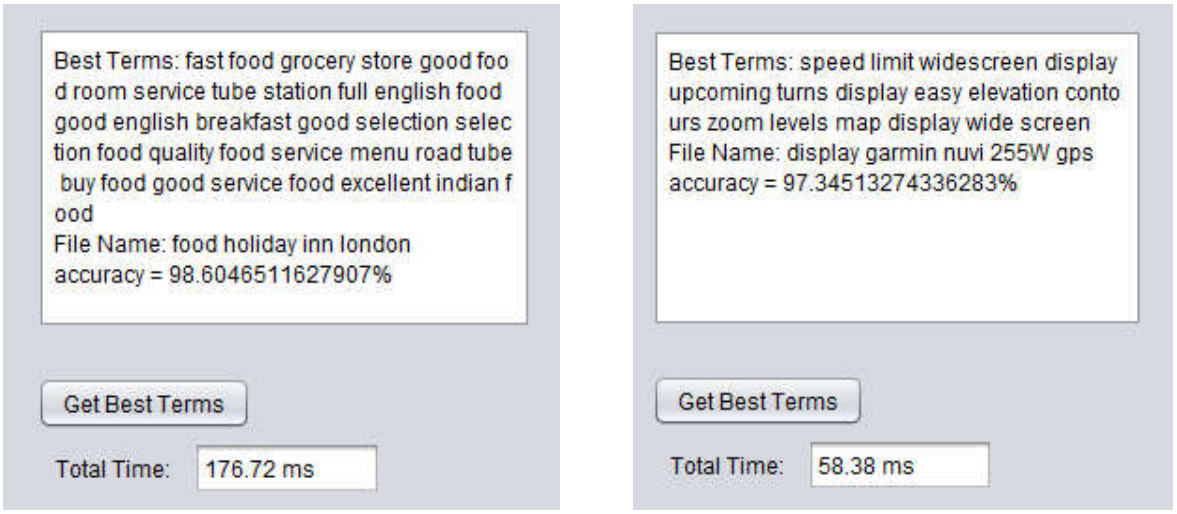


Figure(4.24): An Example of the Most Frequent Terms of a Document for Various Threshold Values and Features Types for Dataset2.

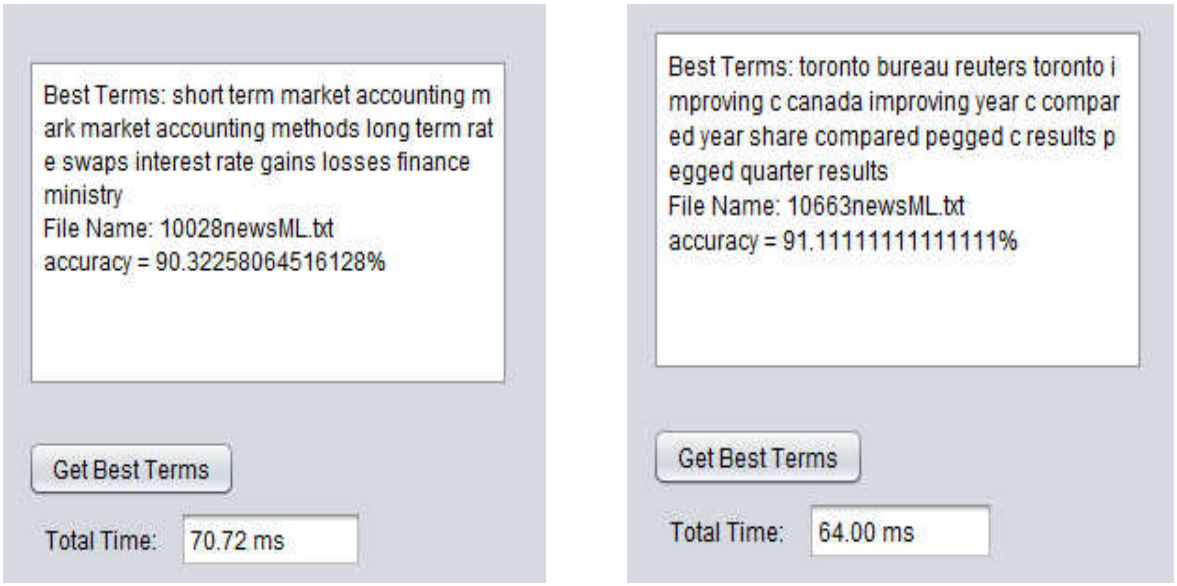
From figures(4.23) and (4.24), the probability for each two terms that may appear in the document is calculated to be an input to the next step.

4.6.2 Applying Levenshtein Distance algorithm

After deploy pattern step, the Levenshtein distance which is described previously in equation (2.2) is applied on the resulting terms from deploy pattern step, the accuracy of each document can be acquired by implement equation (3.1). Figure(4.25) states calculating the accuracy of a document when the threshold is 2 for relative support of dataset1. Figure (4.26) states calculating the accuracy of a document when the threshold is 4 for absolute support of dataset2.



Figure(4.25): The Calculation of the Accuracy of Two Documents when the Threshold is 2 for Relative Support of Dataset1.

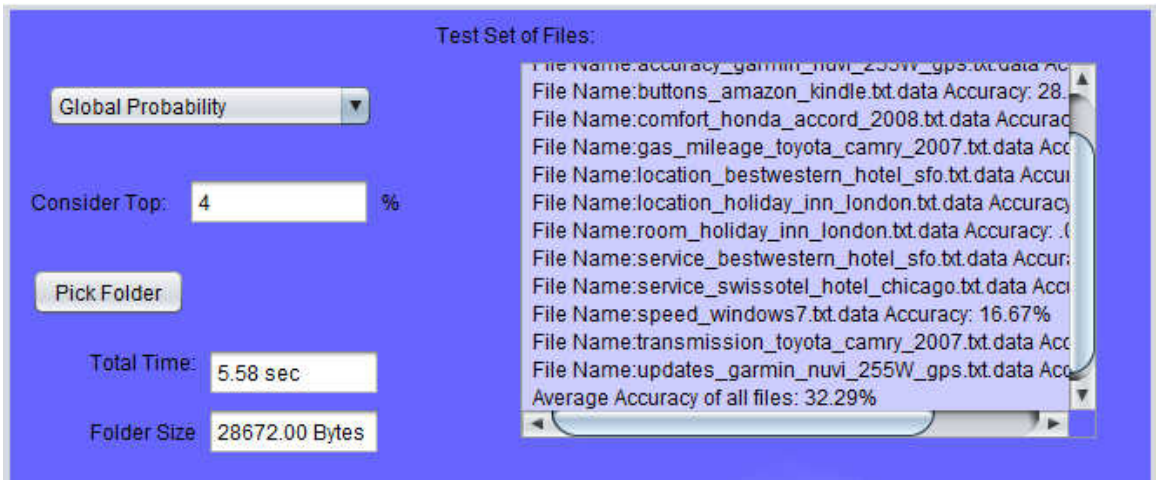


Figure(4.26): The Calculation of the Accuracy of Two Documents when the Threshold is 4 for Absolute Support of Dataset2.

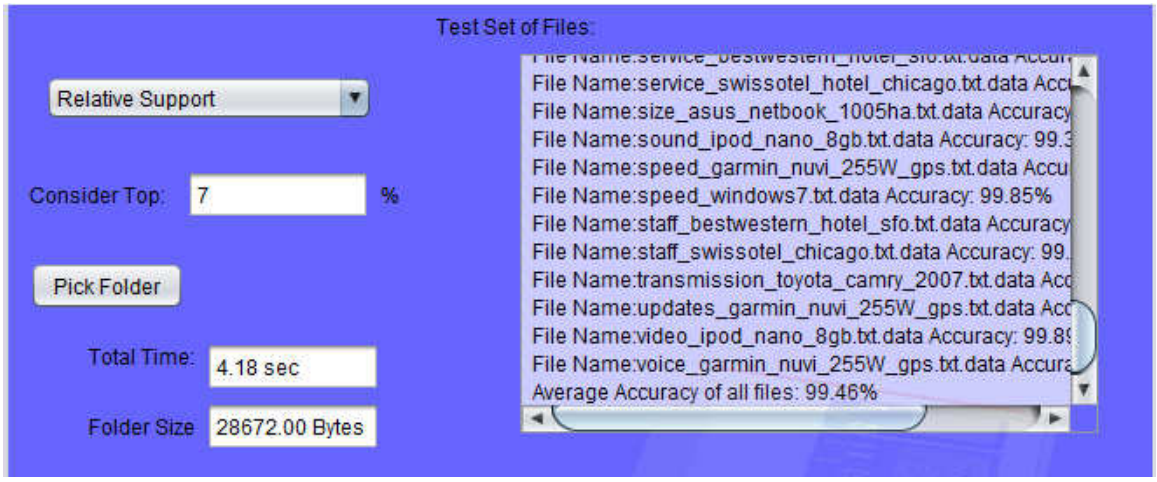
In figures(4.25) and (4.26), levenshtein edit distance algorithm implemented on the resulting terms and the title of the document to find the accuracy of a document by using eq.(2.2).

4.7 The Average accuracy and Time of the dataset for bigram grammar

In this step, after calculating the accuracy of each document in the dataset, equation(3.2) is implemented to compute the average accuracy of a dataset. Figures (4.27) and (4.28) are an examples of calculating the average accuracy of a dataset1 and dataset2 for different threshold values and different features.



(a)



(b)

Figure (4.27): An Example of Calculating the Average Accuracy of a Dataset1 for Bigram Grammar for Different Threshold Values and Different Features,(a) threshold=(4) for global probability,(b) threshold=(7) for relative support.

Figure (4.28): An Example of Calculating the Average Accuracy of a Dataset2 for Bigram Grammar for Different Threshold Values and Different Features,(a) threshold=(2) for absolute support,(b) threshold=(8) for global probability.

From figures (4.27) and (4.28), the proposed system find the average accuracy of a set of documents by finding the sum of accuracy for each document and divide the result by the number of documents .

4.7.1 Results by Average Accuracy and Time for Dataset1

In this subsection, the results of finding the average accuracy of the proposed system for dataset1 based on the value of the threshold for one of the

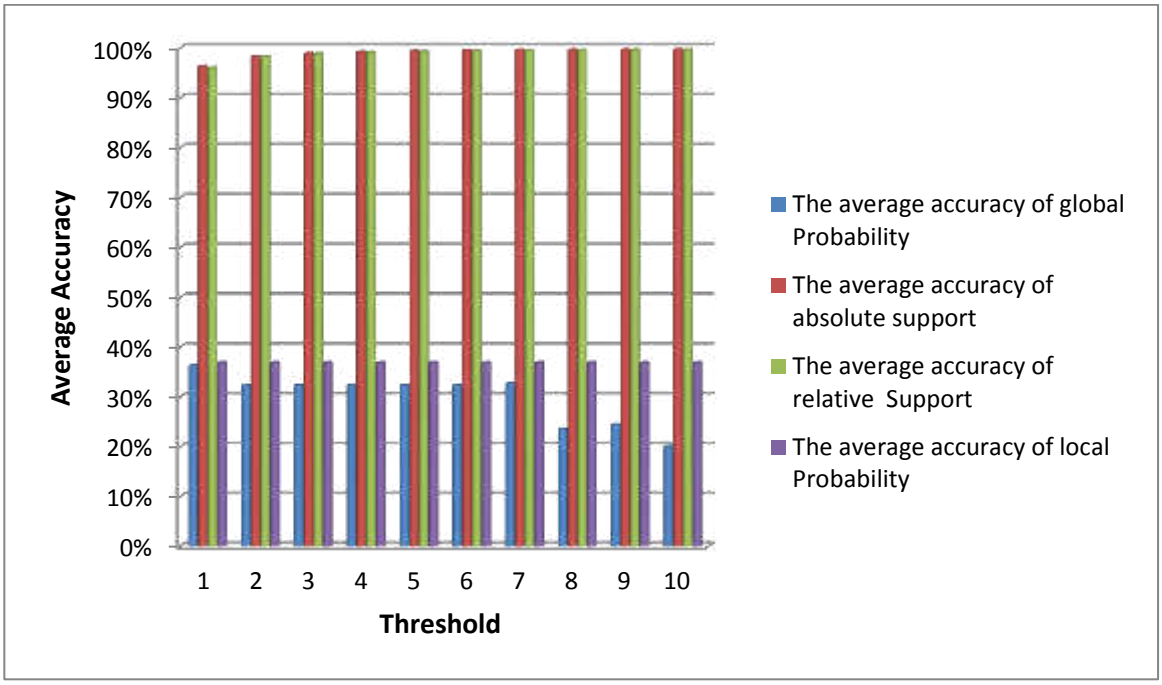
four features are shown. From the experiments on the proposed system, we get that the relative support is equal to covering set for each text. Table (4.3) describes in details the average accuracy for the dataset1of bigram grammar.

Table(4.3): The Average Accuracy and Time of the Proposed System on the Dataset1for Bigram Grammar.

Threshold	The average accuracy of global Probability	Elapse Time In sec	The Average accuracy of absolute support	Elapse Time In sec	The average accuracy of relative Support	Elapse Time In sec	The average accuracy of local Probability	Elapse Time In sec
1	36.21%	3.41	96.29%	4.03	95.99%	3.31	36.9%	3.01
2	32.29%	3.03	98.24%	3.24	98.13%	3.26	36.9%	3.08
3	32.29%	3.05	98.84%	3.22	98.8%	3.37	36.9%	3.24
4	32.29%	3.11	99.16%	3.25	99.1%	3.24	36.9%	3.06
5	32.29%	3.01	99.33%	3.24	99.25%	3.23	36.9%	3.24
6	32.29%	2.98	99.46%	3.36	99.37%	3.33	36.9%	3.05
7	32.62%	3.05	99.52%	3.35	99.46%	3.25	36.9%	3.14
8	23.54%	3.09	99.58%	3.32	99.52%	3.24	36.9%	3.17
9	24.4%	3.02	99.62%	3.31	99.57%	3.32	36.9%	3.01
10	20.04%	3.10	99.65%	3.29	99.61%	3.28	36.9%	3.04

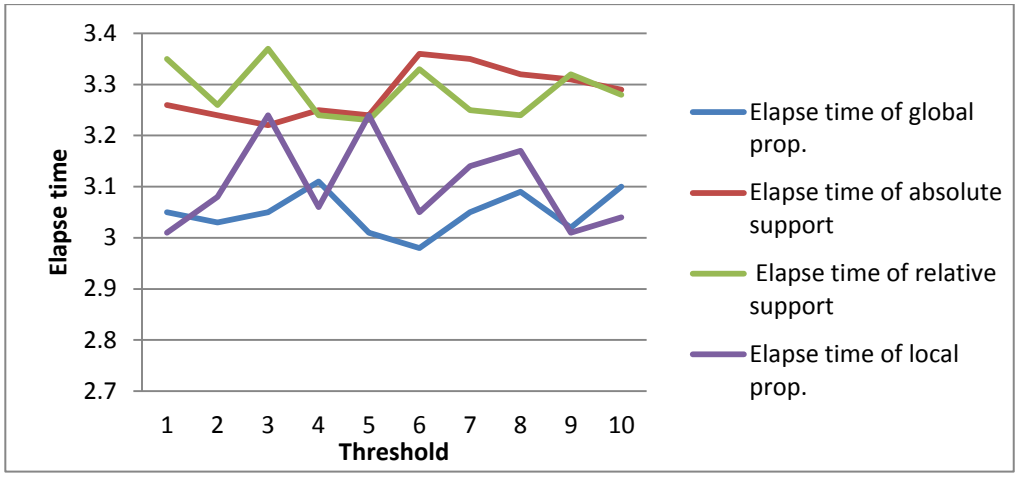
From table (4.3), the average accuracy of global probability for threshold values from 1 to 10 are decreasing from 36.21% to 20.04% because the probability of the term appearing in the document decreases as the threshold value increases. The average accuracy of absolute support for threshold values from 1 to 10 are increasing from 96.29% to 99.65% because the appearing of term in the paragraph increases as the threshold value increases, and the average accuracy of relative support for threshold values from 1 to 10 are increasing from 95.99% to 99.61% because the appearing of term in the fraction of paragraph increases as the threshold value increases. Through testing, the proposed system got that the highest average accuracy which is

99.65% is when the threshold value is 10 for absolute support. Table (4.3) also shows that time is convergent and decreases slightly with an increased threshold. Figure(4.28) states the relationship between threshold values and features values for bigram grammar for dataset1.

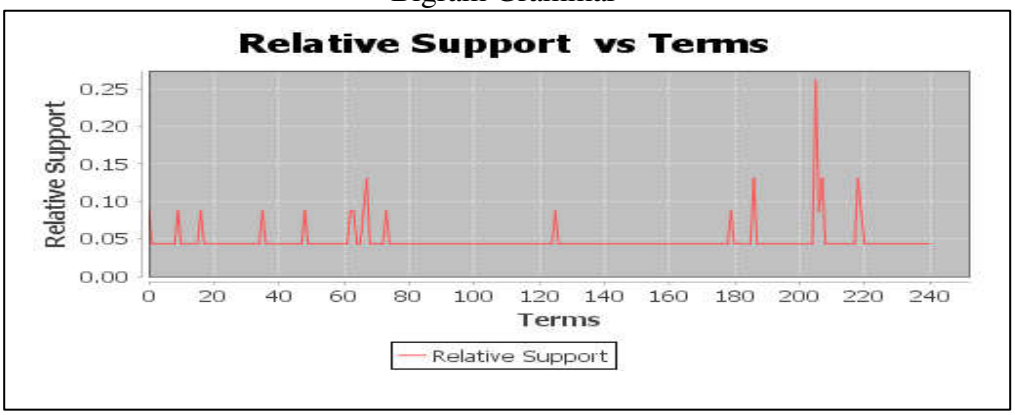


Figure(4.29): The Relationship between Threshold Values and Features Values for Bigram Grammar for Dataset1.

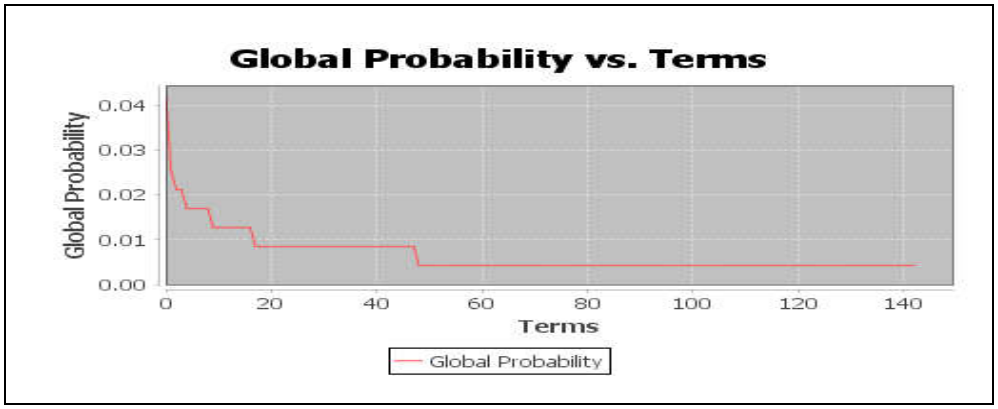
From figure (4.29),When the value of threshold=10, the proposed system acquire the highest average accuracy for absolute support and relative support of a set of files . Figure(4.30) shows the relationship between threshold values and elapse time. Figures ((4.31), (4.32), (4.33), and (4.34)) sequentially states graph representation for different threshold values for bigram grammar. Figures (4.31) and (4.34) proves that the value of relative support and covering set are equal for every threshold value in the proposed system.



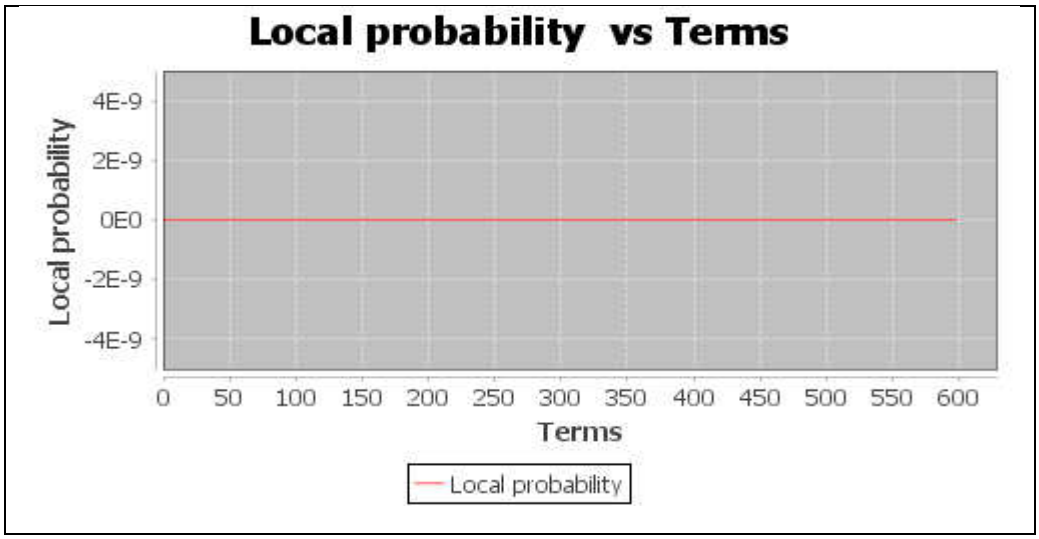
Figure(4.30): The Relationship between Threshold Values and Elapse Time for Bigram Grammar



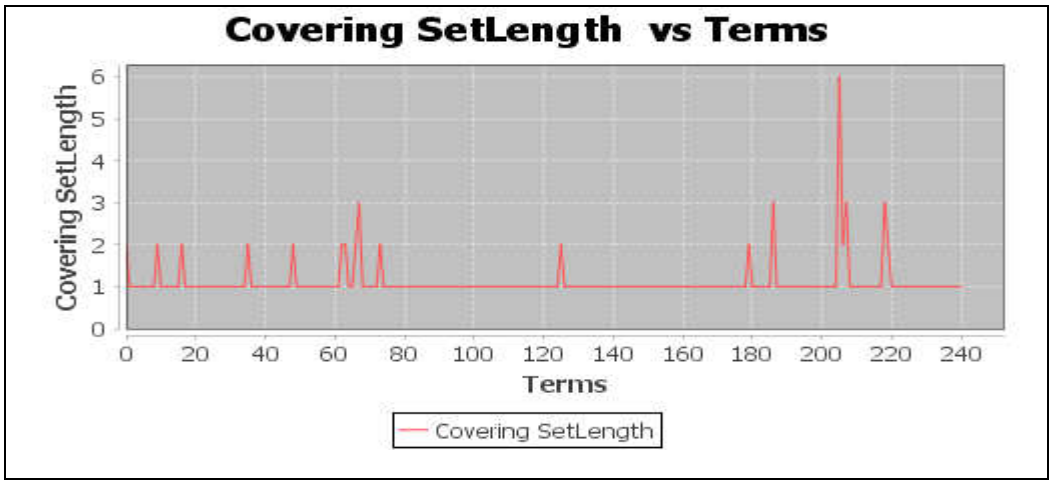
Figure(4.31): An Example of a Graph Representation of Relative Support when Threshold Value =3.



Figure(4.32): An Example of a Graph Representation of Global Probability when Threshold Value =3.



Figure(4.33): An Example of a Graph Representation of Local Probability when Threshold Value =6.



Figure(4.34): An Example of a Graph Representation of a Covering Set when Threshold Value =3.

4.7.2 Results by Average Accuracy and Time for Dataset2

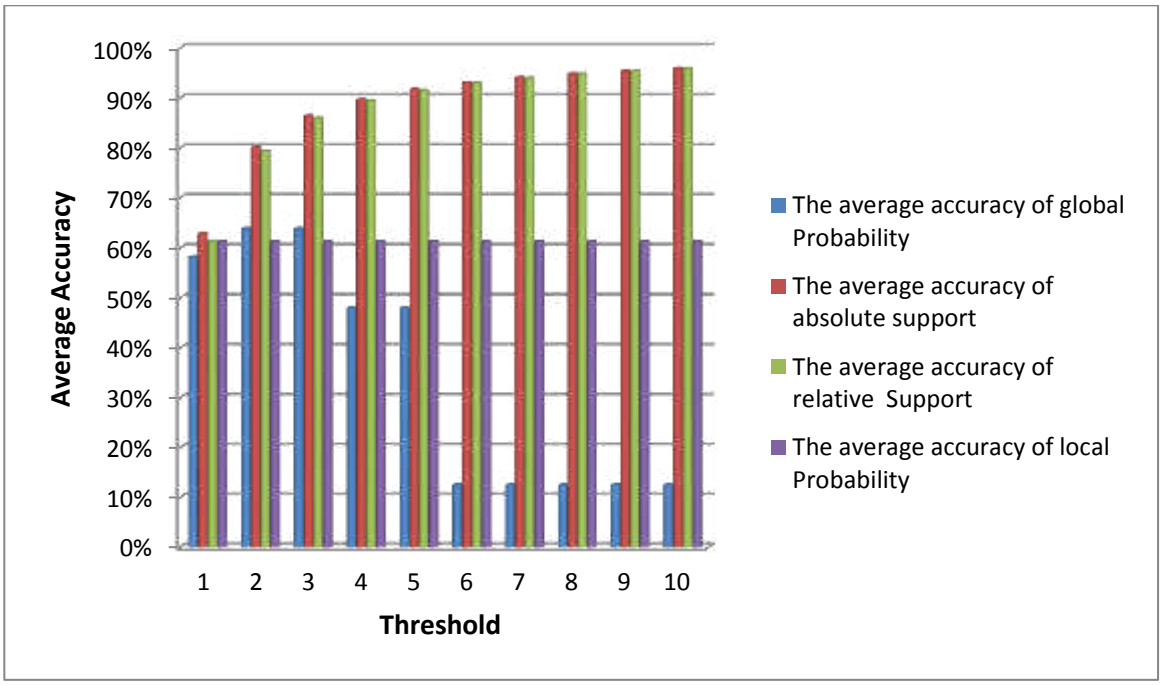
The results of the proposed system on dataset2 for bigram is implemented in this subsection**,** from the experiments on the proposed system, we get that the relative support is equal to covering set for each text. Table (4.4) describes in details the average accuracy for the dataset2.

Table (4.4): The Average Accuracy and Time of the Proposed System on the Dataset2 for Bigram Grammar.

Threshold	The average accuracy of global Probability	Elapse Time In sec	The average accuracy of absolute support	Elapse Time In sec	The average accuracy of relative Support	Elapse Time In sec	The average accuracy of local Probability	Elapse Time In sec
1	58.18%	15.90	62.88%	53.96	61.33%	33.50	61.28%	18.52
2	63.89%	15.63	80.09%	30.64	79.27%	28.80	61.28%	18.37
3	63.89%	16.60	86.42%	37.13	85.98%	31.62	61.28%	19.12
4	48.0%	15.23	89.65%	29.53	89.42%	27.40	61.28%	18.14
5	48.0%	16.26	91.73%	38.65	91.54%	28.45	61.28%	18.82
6	12.5%	15.42	93.06%	38.40	92.96%	29.77	61.28%	18.31
7	12.5%	15.74	94.05%	28.98	93.98%	29.28	61.28%	19.33
8	12.5%	15.43	94.83%	30.36	94.76%	35.91	61.28%	19.29
9	12.5%	15.53	95.4%	31.30	95.35%	37.23	61.28%	19.74
10	12.5%	15.58	95.88%	36.01	95.84%	28.45	61.28%	18.75

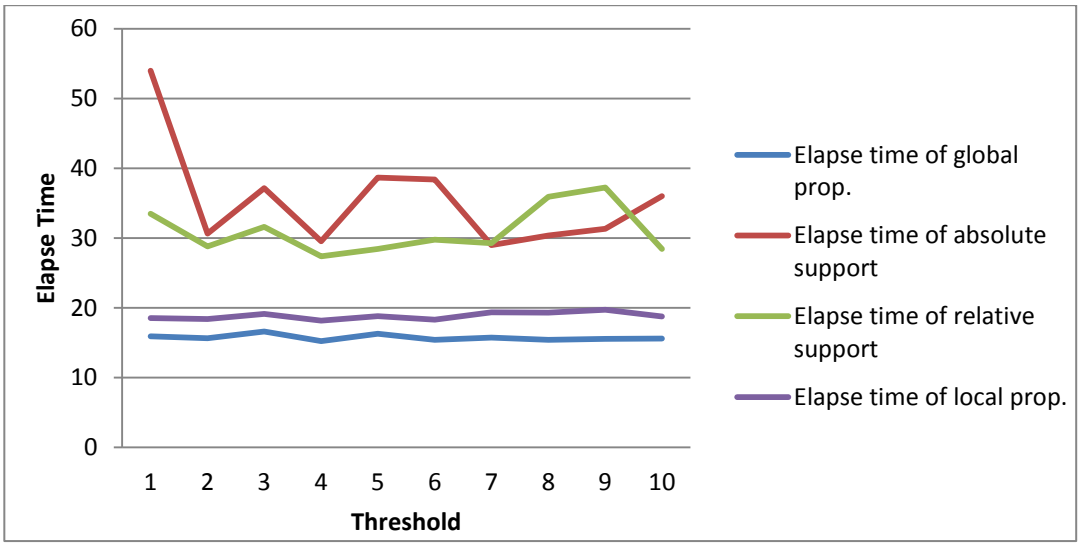
From the experiments of the proposed system for bigram grammar of dataset2, the average accuracy of global probability for the threshold value from 1 to 10 are decreasing from 58.18% to 12.5% because the probability of a term appearing in the document decreases as the threshold value increases. The average accuracy of absolute support for threshold values from 1 to 10 are increasing from 62.88% to 95.88% because the appearing of term in the paragraph increases as the threshold value increases, and the average accuracy of relative support for threshold values from 1 to 10 are increasing from 61.33% to 95.84% because the appearing of term in the fraction of paragraph increases as the threshold value increases. Local probability values for threshold values from 1 to 10 are fixed because the number of bigram terms are equal for every threshold value.

Table (4.2) shows that time is convergent and decreases slightly with an increased threshold. From the experiments on the dataset2, the proposed system obtained a higher average accuracy when the threshold value is 10 for relative support. Figure(4.35) states the relationship between threshold values and features values for dataset2.



Figure(4.35): The Relationship between Threshold Values and Features Values for Dataset2.

In figure (4.35), when the value of threshold=10, the proposed system has the highest average accuracy for absolute support and relative support. Figure(4.36) state the relationship between threshold values and elapse time. Figures ((4.37), (4.38), and (4.39)) sequentially state graph representation for different threshold values. Figures (4.38) and (4.39) prove that the value of relative support and covering set are equal for every threshold value.



Figure(4.36): The Relationship between Threshold Values and Elapse Time

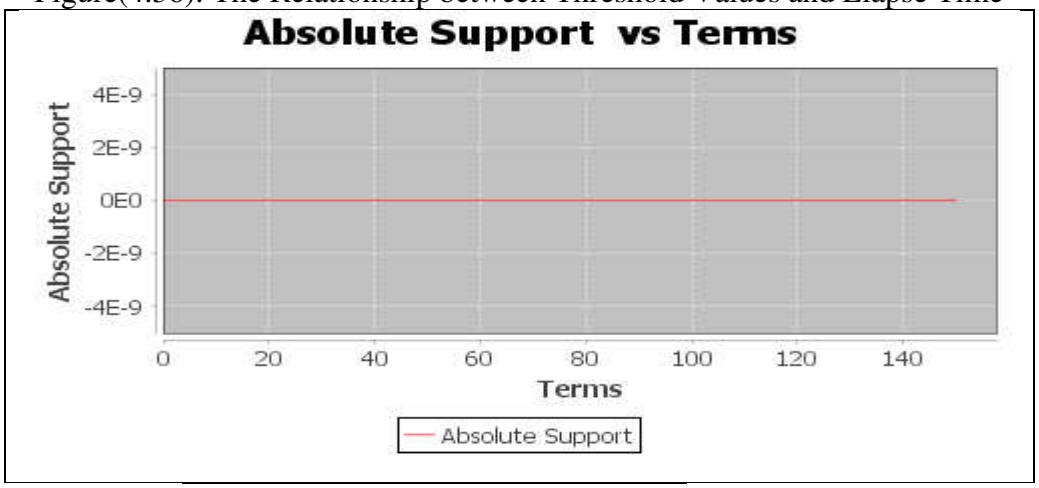


Figure (4.37): An Example of a Graph Representation for Absolute Support when Threshold Value=5

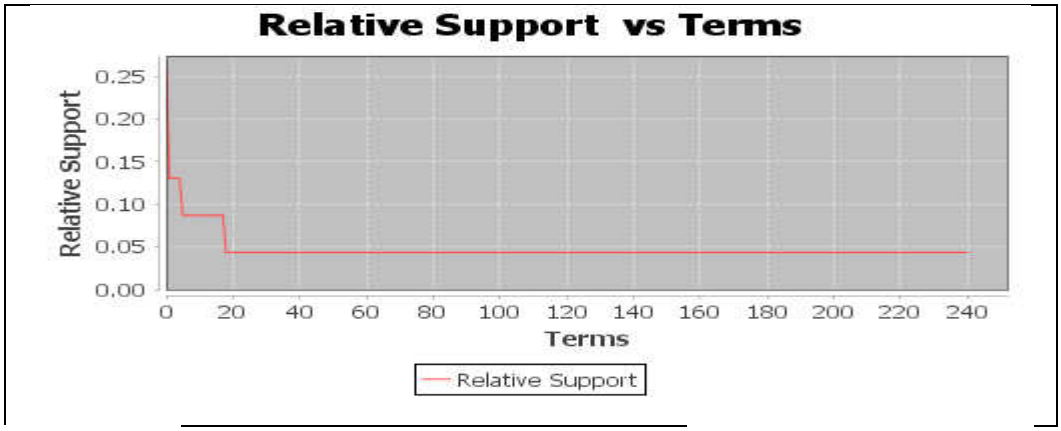


Figure (4.38): An Example of a Graph Representation for Relative Support when Threshold Value = 4.

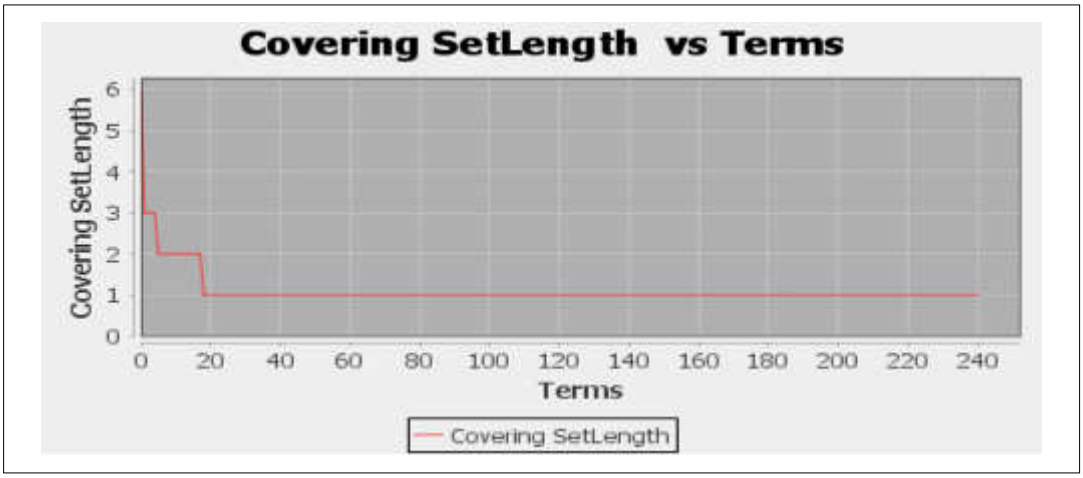


Figure (4.39): An Example of a Graph Representation for Covering Set when Threshold Value = 4.

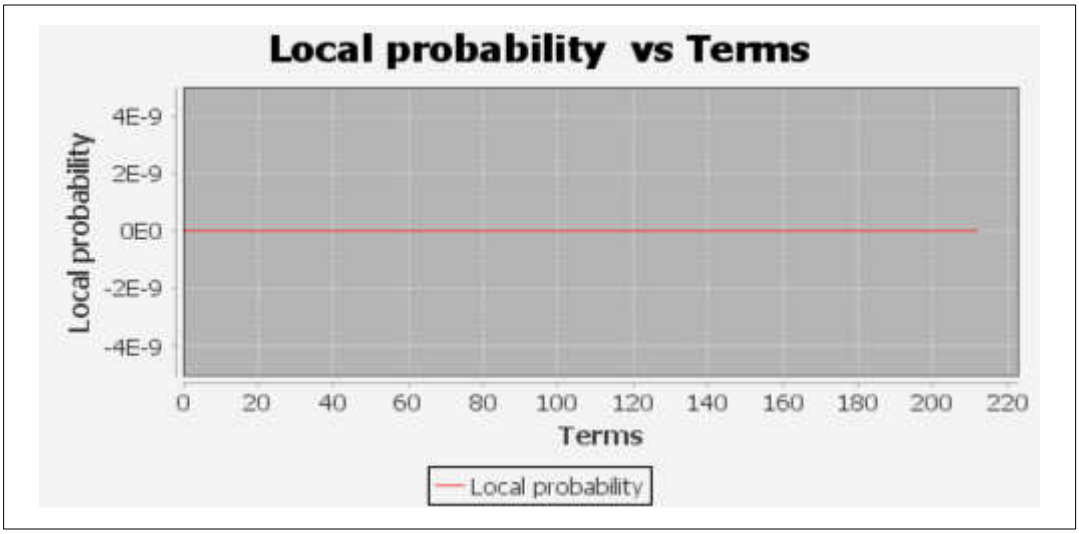


Figure (4.40): An Example of a Graph Representation for Local Probability when Threshold Value = 1.

4.8 Comparison between our proposed system and other existing systems

The comparison results of the proposed system with other systems that worked to extract a pattern from a specific document or text are explained in the following:

1- A system by H. M. M. Hasan which designed to extract keywords from documents by applying modified semantic relation approach which gets an accuracy of 77.6% precision and 84.3% recall to measure the accuracy of the system.

2- A system by V. Aswini that used the process of pattern evolving and pattern deploying which get 62% of precision and 82% of recall.

3- A system by D. P. Lyras that used the Levenshtein Edit Distance on a dictionary-based algorithm for automatic lemmatization for modern English which achieved an accuracy 96% for English language.

In the systems described above, they used different standards to measure the efficiency of the patterns extracted from the text. In the proposed system, different method is use, average accuracy is used to measure the efficiency of the system more clearly. The proposed system achieved results for Uni-gram grammar (98.68% average accuracy for absolute support of dataset1 and 88.89% average accuracy for relative support of dataset2) and for Bi-gram grammar, it get (99.65% average accuracy for absolute support of dataset1 and 95.88% average accuracy for absolute support of dataset2). So, the results of the proposed system are higher than the previous systems which designed for discovering pattern from text document.

Chapter Five

Conclusions & Suggestions for Future Works

Chapter Five

Conclusions & Suggestions for Future Works

5.1 Conclusions

In this thesis, we have mainly focused on developing mining algorithm for discovering patterns from a large data collection . There are some conclusions that were obtained through experiments conducted on the proposed system and in the following some of these conclusions are listed:

1. Text mining is similar to data mining in terms of dealing with large volumes of data, and both fall into the information discovery area also both of text mining and data mining uses the same techniques.
2. Pattern-based method can be utilized for finding different patterns for text without any problems.
3. The research provides a technique for discovering patterns by using (LDA) with (PTM) for improving the effectiveness of the system and gives best results compared to other methods.
4. The experimental results of the proposed system for relative support is equal to the experimental results of the covering set.
5. Threshold value that gives the system a highest average accuracy is 10, When the threshold value exceeds 10, It will approach the size of the document as a whole, affecting the value of the information extracted from the text.
6. From the experimental results, the value of local probability had very little effect on the results because it means the probability of term in the paragraph not in the document.

7. As a result of use LDA and PTM algorithms, the proposed system was got 98.68% of average accuracy for absolute support of dataset1 and 88.90% of average accuracy for relative support of dataset2 with a very short time of processing for unigram grammar.
8. The proposed system obtained 99.65% of average accuracy for absolute support of dataset1 and 95.88% of average accuracy for relative support of dataset2 as a result with a very short time. These results were better than other systems for pattern discovery in text mining.
9. Bigram results better than unigram because bigram treats each two words as one unit and it reduce the size of a document. It is useful in terms of the linguistic structure of the sentences especially if the words are name and adjective, while unigram takes one word and associating it with another word may not give correct language.

5.2 Suggestion for Future Works

Several suggestions can be placed to develop this work such as:

1. Extract the remainder N-gram such us tri-grams and quad-grams.
2. Implement a neural network approaches with our algorithm to improve the results.
3. Train Hidden Markov Model (HMM) to extract patterns from documents and update the discovered patterns to find interesting and relevant information.
4. The proposed system can be personalized, so that data or results obtained from the text document is concerned with the profile of the user. Here, to search for information, the profile of the user will be generated to the system. After that, the documents that match the user profile will be extracted.

References

References

[1] B. Inje and U. Patil, **" Operational Pattern Revealing Technique in Text Mining"**, *2014 IEEE Students Conference on Electrical, Electronics and Computer Science*, Bhopal, pp.1-5, 2014.

[2] S. M. Ali and Prof. Ms. R.R.Tuteja, **"Data Mining Techniques"**, *International Journal of Computer Science and Mobile Computing(IJCSMC)*, Vol. 3, Issue:4, pp.879-883, April 2014.

[3] M. Suganthy, K. Rupika and J. S. Fransuva, **"Text Mining for Pattern Identification"**, *International Journal of Futuristic Science Engineering and Technology*, ISSN:2320–4486, Vol.1, Issue:3, March2013.

[4] S. M .Inzalkar and J .Sharma, **"A Survey on Text Mining- Techniques and Application"**, International Journal of Research In Science & Engineering, Special Issue: Techno-Xtreme 16, e-ISSN: 2394-8299, p-ISSN:2394-8280.

[5] A. G. Jivani, **"A Comparative Study of Stemming Algorithms"**,[Online].Available at: http//www.ijcta.com, ISSN:2229-6093, Vol.2, pp.1930-1938, Nov 2011.

[6] D.T .Larose and C.D .Larose, **"Discovering Knowledge in Data: An Introduction to Data Mining"**, Book from John Wiley & Sons Publishing 2014, ISBN 0-471-66657-2, 2014.

[7] N. Zhong, Y. Li and S. Wu, **"Effective Pattern Discovery for Text Mining",** in *IEEE Transactions on Knowledge and Data Engineering*, Vol.24, No.1, pp. 30-44, Jan. 2012.

[8] D. S. Charjan and Prof. M. A. Pund**, "Pattern Discovery For Text Mining Using Pattern Taxonomy"**, *International Journal of Engineering Trends and Technology (IJETT)*, ISSN:2231-2803, Vol.4, Issue 10, pp.4550-4555, October 2013.

[9] B. Laxman and D. Sujatha, **"Improved Method for Pattern Discovery in Text Mining"**, *International Journal of Research in Engineering and Technology (IJRET)*, e-ISSN: 2319-1163, p-ISSN: 2321-7308, Vol.02, Issue 10, Oct 2013.

[10] C. Kadu, P. Bhanodia and P. Jain, **"Hybrid Approach to Improve Pattern Discovery in Text mining",** *International Journal of Advanced Research in Computer and Communication Engineering***,** Vol. 2, Issue:6, June 2013.

[11] V.Aswini and S.K.Lavanya, **"Pattern Discovery for Text Mining"**, *2014 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC),* Chennai, pp.412-416, 2014.

[12] M.R. Shinde and P .C. Gill, **"Pattern Discovery Techniques for the Text Mining and its Applications**", *International Journal of Science and Research (IJSR)*, ISSN:2319-7064, Impact Factor (2012):3.358, Vol.3, Issue:5, May 2014.

[13] V. Pansare, **" Effective Pattern Identification Approach for Text Mining"**, *International Journal of Computer Science and Information Technologies (IJCSIT)*, ISSN:0975-9646, Vol.7, pp.1826-1830, 2016.

[14] S. R. Lomate, **"Advanced A priori Algorithm for Effective Pattern Discovery in Text Mining"**, *International Journal of Engineering Science and Computing (IJESC)*, ISSN:2321- 3361, Vol.6, Issue:6, pp.7654-7661, June 2016.

[15] H. M. M. Hasan, F. Sanyal and D. Chaki, **"A Novel Approach to Extract Important Keywords from Documents Applying Latent Semantic Analysis"**, *2018 10th International Conference on Knowledge and Smart Technology (KST),* Chiang Mai, pp. 117-122, 2018.

[16] V. Gupta and G.S. Lehal, **"A Survey of Text Mining Techniques and Applications"**, *Journal of Emerging Technologies in Web Intelligence*, Vol.1, No.1, pp.4550-4555, August 2009.

[17] S. D. Gupta and B. P. Vasgi, **" Implementation of pattern discovery to retrieve relevant document using text mining"**,*2015 International Conference on Green Computing and Internet of Things (ICGCIoT),* Noida, pp.327-332, 2015.

[18] B. Shankaran, M. Patil, S. Suryawanshi, S. Mandhane and S. S. Raskar, **"A Novel Approach for Text Extraction using Effective Pattern Matching Technique"**, *International Journal of Research in Engineering and Technology (IJRET)*, e-ISSN: 2319-1163, p-ISSN: 2321-7308, Vol.04, Issue:01, pp.269-272, Jan 2015.

[19] M. Sukanya and S. Biruntha, **"Techniques on Text Mining"**, *2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram, pp.269-271, 2012.

[20] D. Nasa, **"Text Mining Techniques - A Survey "**, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, Vol.2, Issue:4, pp. 51-540, April 2012.

[21] D. J. G. Sundari and D. Sundar, **"A Study of Various Text Mining Techniques"**, *International Journal of Advanced Networking & Applications (IJANA)*, VoL.08, Issue:05, pp.82-85, 2017.

[22] S. Dang and P.H. Ahmad, **"Text Mining: Techniques and its Application"**,

IJETI International Journal of Engineering & Technology Innovations, ISSN(Online):2348-0866, Vol.1, Issue:4, pp.22-25, Nov 2014.

[23] P. Monali and K. Sandip, **"A Concise Survey on Text Data Mining"**, *in Proceeding of the International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 3, Issue:9, pp.8040- 8043, 2014.

[24] R. Balamurugan and Dr. S. Pushpa, **"A Review on Various Text Mining Techniques and Algorithms "**, *2nd International Conference on Recent Innovation in Science, Engineering and Management (ICRISEM 2015)*, JNU Convention Center, Jawaharlal Nehru University, New Delhi, pp.837-848, 22 Nov 2015.

[25] S.Sheela and T.Bharathi , **"Analyzing Different Approaches of Text Mining Techniques and Applications"**, *International Journal of Computer Science Trends and Technology (IJCST)*, ISSN:2347-8578, Vol.6, Issue:4, pp.23-29, Aug 2018.

[26] S.V. Gaikwad and P. Patil, **"Text Mining Methods and Techniques"**, *International Journal of Computer Applications (0975 – 8887)*, Vol. 85, No.17, January 2014.

[27] A. Mustafa, A. Akbar and A. Sultan, **"Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization"**, *International Journal of Multimedia and Ubiquitous Engineering* ,Vol. 4, No.2, pp.183-188, April 2009.

[28] V. Gupta and G. S. Lehal, **"A Survey of Text Mining Techniques and Applications"**, *Journal of Emerging Technologies in Web Intelligence*, Vol.1, No.1, pp.60-76, August 2009.

[29] R. Sagayam, S. Srinivasan and S. Roshni, **" A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques"**, *International Journal Of Computational Engineering Research (IJCER)*, ISSN:2250-3005, Vol.2, Issue:5, pp.1443-1446, September 2012.

[30] Y. Song, B. Li and R. Acharya, **"Machine Learning Text Mining: Classification, Retrieval and Recommendation"**, Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, The Pennsylvania State University, May 2009.

[31] Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, **"IKNN: Informative K-Nearest Neighbor Pattern Classification"**, In: Kok J. Koronacki, R.L. de Mantaras , S. Matwin, D. Mladenic, A. Skowron, **"Knowledge Discovery in Databases: PKDD 2007"**, Lecture Notes in Computer Science, Vol.4702. Springer, Berlin, Heidelberg, pp. 248–264, 2007.

[32] K. Mythili and K. Yasodha, **"A Pattern Taxonomy Model with New Pattern Discovery Model for Text Mining "**, *International Journal of Science and Applied Information Technology (IJSAIT)*, ISSN:2278-3083,Vol.1, No.3, pp.88-92, July 2012.

[33] J. Yang, X. Yang and J. Zhang, **"A Parallel Multi-Class Classification Support Vector Machine Based on Sequential Minimal Optimization",** *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, Hangzhou, Zhejiang, pp.443-446, 2006.

[34] F. Kyoomarsi, A. Tajoddin, P. Dehkordy, H. Khosravi and E. Eslami **"Optimizing Text Summarization Based on Fuzzy Logic"**, *in 2008 7th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2008)*, Portland, OR, pp. 347-352, 2008.

[35] Y. Yang, J. Zhang, and B. Kisiel. **"A Scalability Analysis of Classifiers in Text Categorization"**. *In SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, USA, pp. 96–103, 2003.

[36] D.J.G. Sundari and D. Sunder, **"A Study of Various Text Mining Techniques"**, *International Journal of Advanced Networking &Applications(JIANA)*, Vol. 08, Issue:05, pp.82-85, 2017.

[37] D.C.Sonawane, T.P.Shirole, K.D.Patil, P.V. Patil, A.K. Patil, **" Effective Pattern Discovery for Text Mining",** *International Research Journal of Engineering and Technology (IRJET)*, Vol.04, Issue:04, pp.194-197, 2017.

[38] F. S. Gharehchopogh and Z. A. Khalifelu, **"Analysis and Evaluation of Unstructured Data: Text Mining Versus Natural Language Processing "**, *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*, Baku, pp.1-4, 2011.

[39] P.K. Jayasekara, K.S.Abu , **"Text Mining of Highly Cited Publications in Data Mining",** 5th International Symposium on Emerging Trends and Technology in Libraries and Information Services, IEEE, pp.128-130, 2018.

[40] A. Akilan, **"Text Mining: Challenges and Future Directions**", IEEE *2nd International Conference on Electronics and Communication Systems 2015(ICECS)*, Coimbatore, pp.1679-1684, 2015.

[41] M.F. Porter, **"An Algorithm for Suffix Stripping"**, Program: Electronic Library and Information Systems, Emerald Group Publishing Limited, Vol.40, No.3, pp.211-218, 2006.

[42] Porter and M.F., **"Stemming Algorithms for Various European Languages"**,[Online]. available at: www.snowball.tartarus.org/texts/ stemmers overview, accessed 18 April 2006 .

[43] D. A. Hull, **"Stemming Algorithms-A Case Study for Detailed Evaluation"**. Journal of the American Society for Information Science, pp.70-84, 2006.

[44] C. Moral, A.D. Antonio, R. Imbert and J. Ramirez, **"A survey of stemming algorithms in information retrieval"**, Information research, Vol.19, No.1, March 2014.

[45] W. Ben Abdessalem Karaa, **"A New Stemmer to Improve Information Retrieval "**, *International Journal of Network Security & Its Applications (IJNSA*), Vol.5, No.4, pp.143-145, July 2013.

[46] D. P. Lyras, K. N. Sgarbas and N. D. Fakotakis, *"***Using the Levenshtein Edit Distance for Automatic Lemmatization: A Case Study for Modern Greek and English**",*19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, Patras, pp.428-435, 2007.

[47] A. Niewiarowski, **"Short Text Similarity Algorithm Based on The Edit Distance and Thesaurus"**, Technical Transactions, Fundamental Sciences, pp.160-170, 2016.

[48] S-S. Kang, **"Word Similarity Calculation by Using the Edit Distance Metrics with Consonant Normalization"**, *Journal of Information Processing Systems (JIPS)*, Vol.11, No.4, pp.573-582, December 2015.

[49] P. A. Utama and B. Distiawan, **" Spark-Gram: Mining Frequent N-grams using Parallel Processing in Spark",** *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Depok, pp. 129-136, 2015.

[50] William B. Cavnar and John M. Trenkle, **"N-Gram-Based Text Categorization"**, In Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, 2004.

[51] J. Houvardas and E. Stamatatos, **" N-gram Feature Selection for Authorship Identification ",** in Proc. of the 12th Int. Conf. on Artificial Intelligence: Methodology, Systems, Applications, Vol.4183, Springer, Berlin, Heidelberg, pp.77-86, 12-15 September 2006.

[52] P. McIlroy, **"Optimistic Sorting and Information Theoretic Complexity ",** Proceeding SODA '93 Proceedings of the fourth annual ACM-SIAM symposium on Discrete algorithms, Austin, Texas, USA, pp. 467-474, 25 – 27 January 1993.

[53] K. Ganesan, C. Zhai and J. Han., **" Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions"**, *In Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China, pp.340-348, 2010.

[54] K. Ganesan, **Opinosis Opinion**, Beijing, China, 2010, Accessed on: Jun 12, 2019, [Online]. Available :https//archive.ics.uci.edu/ml/Datasets.

[55] ZhiLiu, **Reuter_50_50**, Hubei Wuhan, China, 2011, Accessed on: March 2,2019, [Online].Available at: https//archive.ics.uci.edu/ml/

Datasets/Reuters/Reuter _50_50.

وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كلية العلوم
قسم علوم الحاسوب

اكتشاف النمط لتعدين النص يقاس بواسطة مسافة ليفنشتين

رسالة

مقدمة الى قسم علوم الحاسوب/كلية العلوم /جامعة ديالى كجزء من متطلبات نيل درجة الماجستير في اختصاص علوم الحاسوب

من قبـــل
ليلى عبد الحق اسماعيل
بإشراف
أ.ناجي مطر سحيب

ايلول ٢٠١٩

الخلاصة

كمية المعلومات النصية تتراكم بسرعة والتي يتم تخزينها إلكترونيًا على أجهزة الكمبيوتر أو الويب. أي جهاز كمبيوتر (كمبيوتر محمول أو سطح مكتب) قادر على استيعاب كميات هائلة من البيانات بسبب التحسينات في أجهزة التخزين.

يتم تضمين النصوص في مجموعة بيانات نصية ومجموعة البيانات هذه تكون غير مهيكلة. يمكن معالجة هذه البيانات غير المهيكلة عن طريق تعدين النص. يكشف التعقيد والعدد الكبير لهذه البيانات عن العديد من القدرات الجديدة للمحللين. لذلك ، يقدم هذا العمل تحسينًا لاستخراج الأنماط المفيدة من المستندات النصية في مجال تعدين النصوص باستخدام نموذج تصنيف الأنماط (PTM) وخوارزمية مسافة Levenshtein (LDA).

هناك طرق مختلفة للتعامل مع المستندات النصية. في هذه الأطروحة ، تم اقتراح نظام لتعدين النص للتغلب على المشاكل التي حدثت في الطريقة القائمة على المصطلح والطريقة القائمة على العبارة. يعتمد النظام المقترح على سلوك خوارزمية LDA و PTM لتحديد أفضل دقة للأنماط المستخرجة في وقت قصير ولإثبات أن الطريقة القائمة على النمط هي الحل الأفضل لتعدين النص دون أي مشاكل في المعلومات المستخرجة من النص.

تم اختبار قوة الخوارزميتين (PTM، LDA) باستخدام قيم العتبة من ١ إلى ١٠ للحصول على ١٪ إلى ١٠٪ من المعلومات في النص. استخدم النظام المقترح "مجموعة بيانات الرأي المفتوح" و "مجموعة بيانات رويترز ٥٠_٥٠" المخزنة في ملف ".txt" أو مستند نصي. تم الحصول على نتائج هذا الاختبار من خلال المقارنة بين قيم اربع ميزات وهي (الاحتمال العالمي، الاحتمال المحلي ، الدعم المطلق ، الدعم النسبي) للنص للحصول على متوسط دقة أعلى.

تم مقارنة نتائج النظام المقترح مع الأنظمة الأخرى. حصل النظام المقترح على متوسط دقة (٩٨.٦٨٪) لقواعد Unigram و (٩٩.٦٥٪) متوسط دقة لقواعد Bigram بينما حقق النظام الذي يستخدم Levenshtein Edit Distance للتحكم التلقائي في اللغة الإنجليزية الحديثة دقة ٩٦٪ للغة الإنجليزية ونظامًا يستخدم عملية تطور النمط ونشر النمط حصل على ٦٢٪ من الدقة و ٨٢٪ من الاستدعاء. لذا ، فإن استخدام LDA مع PTM حقق نتائج أفضل مقارنة بالنظم الأخرى.