THE EFFECTIVENESS OF FEATURE SELECTION METRICS ON THE TEXT CATEGORIZATION PERFORMANCE

ASMAA AL-GARTANEE

JANUARY 2015

THE EFFECTIVENESS OF FEATURE SELECTION METRICS ON THE TEXT CATEGORIZATION PERFORMANCE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY
ASMAA AL-GARTANEE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
INFORMATION TECHNOLOGY PROGRAM

JANUARY 2015

Title of the Thesis : **The Effectiveness of Feature Selection Metrics on the Text Categorization Performance.**

Submitted by **Asmaa Muhamed Aubaid AL-GARTANEE**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Billur K

Prof. Dr. Billur KAYMAKÇALAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist Prof. Dr. Abdül Kadir GÖRÜR
Supervisor

Examination Date: 30.01.2015

Examining Committee Members

Assist. Prof. Dr. Abdül Kadir GÖRÜR	(Çankaya Univ.)
Assoc. Prof. Dr. H.Hakan MARAŞ	(Çankaya Univ.)
Assoc. Prof. Dr. Fahd JARAD	(THK Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Asmaa, AL-GARTANEE
Signature : Asmaa
Date : 30.01.2015

ABSTRACT

THE EFFECTIVENESS OF FEATURE SELECTION METRICS ON THE TEXT CATEGORIZATION PERFORMANCE

AL-GARTANEE, Asmaa
M.Sc., Department of Mathematics and Computer Science
Information Technology Program
Supervisor: Assist. Prof. Dr. Abdül Kadir GÖRÜR
January 2015, 63 pages

Text Categorization (TC) is an important intelligence information processing technology. This technology has high value in information retrieval, Electronic Governments, information filtering, text databases, digital libraries, and other aspects, but the problem of feature selection is equally or more important than text-categorization. In this thesis, we did our experiments with the help of standard Reuters-21578 dataset, and we discussed many important topics ranging from collecting data, to organizing data and ultimately using the organized data to efficiently conduct tests using the feature selection metrics.The general idea of any feature selection metric is to determine importance of words using some measure that can keep informative words, and remove non-informative words, which can then help the text-categorization engine categorize a document, D, into some category, C.

The feature selection metrics that will be discussed in this thesis are: Term frequency-Inverse Document Frequency *(*TF-IDF*)*, Document Frequency (DF), Mutual Information- Explanation (MI), Chi-square Statistics (CHI), GSS (Galavotti-Sebastiani-Simi) Coefficient – Explanation.

It will combine Term frequency-inverse document frequency (TF-IDF) and Documents Frequency (DF) metrics to prepare the texts in a perfect way. After that, those texts will be used by classification process in Weka to get the best learning machines algorithms and the best performance of system, by computing performance measures such as (accuracy, error rate, recall, precision and F-measure). We compare the reusability of popular active learning algorithms for text classification and identify the best classifiers to use in active learning for text classification. All these mentioned measures were computed and plotted.

Keywords: Text Categorization, Feature Selection Metrics, Learning Machines.

ÖZ

ÖZELLİK BELİRLEME MATRİKSİNİN METİN SINIFLANDIRMA SİSTEMİNİN PERFORMANSI ÜZERİNDEKİ ETKİSİ

AL-GARTANEE, Asmaa

Yüksek Lisans, Matematik-Bilgisayar Anabilim Dalı.
Bilgi Teknolojileri Bölümü
Tez Yöneticisi: Yrd. Doç.Dr. Abdül Kadir GÖRÜR
Ocak 2015, 63 sayfa

Metin Sınıflandırma (TC) önemli bir istihbarat bilgi işlem teknolojisidir. Bu teknoloji, bilgi alma, E-devlet, bilgi filtreleme, metin veritabanları, dijital kütüphaneler ve benzeri konularda çok yüksek bir değere sahiptir. Ancak, özellik belirleme konusu, metin sınıflamasından çok daha önemlidir. Bu tezde, biz standart Reuters-21578 veri kümesi ile deneyler yaptık ve veri toplamadan veri organize etmeye varıncaya kadar birçok konuyu irdeledik ve sonunda organize edilmiş verileri kullanarak, özellik belirleme matriksi esasına göre etkin deneyler yaptık. Özellik belirleme matriksinin genel fikri; bilgi içeren sözcükleri muhafaza ederek, bilgi içermeyen sözcükleri ise dışarı atarak işlem yapan bazı ölçütler kullanarak kelimelerin önemini belirlemektir. Böylece metin sınıflama motoruna bir dokumanı (D dokumanı), bir başka dokumana (C dokumanı) dönüştürüp sınıflandırma noktasında yardımcı olunmaktadır.

Bu tezde ele alınacak özellik seçimi ölçütleri şunlardır: Dönem Frekans -Ters Belge Frekans (TF-IDF), Belge Frekans (DF), Karşılıklı Bilgi-Açıklama (MI), Ki-kare İstatistikleri (CHI), GSS (Galavotti -Sebastiani-Simi) Katsayısı - Açıklama.

Bu mükemmel bir şekilde metinleri hazırlamak için Dönem frekans ters belge frekans (TF-IDF) ve Belgeler Frekans (DF) ölçümleri bir araya getirecektir. Bundan sonra, bu metinler en iyi makine algoritmasını ve en iyi sistem performansını elde etmek için, Doğruluk, Hata Oranı, Hatırlama, Hassasiyet ve F-ölçüsü gibi hesaplama performans ölçütlerini temin etmek için Weka'da sınıflandırma işleminde kullanılacaktır. Bu çalışmada, metin sınıflandırması için popüler aktif öğrenme algoritmalarının tekrar kullanılabilirliklerini karşılaştırdık ve metin sınıflaması için aktif öğrenmede kullanılabilecek en iyi sınıflandırıcıları belirledik. Sözü edilen bütün bu ölçütler hesap edildi ve grafiklerde gösterildi.

Anahtar Kelimeler: Metin Sınıflandırması, Özellik Belirleme Matriksi, Öğrenme Makinaları.

ACKNOWLEDGEMENTS

I am very thankful to Asst. Prof. Dr. Abdül Kadir GÖRÜR for the supervision and grateful for the sincere guidance suggestions, and encouragement throughout the development of this thesis.

I am especially grateful to my highly supportive husband and my adorable children for their care, patience and valuable assistance and support during all this time of my study.

TABLE OF CONTENTS

CHAPTERS:

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

BOD	Bag of Document
C	Categories
CC	Correlation Coefficient
CHI	X^2-Test
D	Documents
DF	Document Frequency
DR	Document Representation
FN	False Negatives
FP	False Positives
GSS	Galavotti-Sebastiani-Simi
IDF	Inverse Document Frequency
IG	Information Gain
K-NN	K-Nearest Neighbor
LLSF	Linear Least-Square Fit
Max	Maximization
MI	Mutual Information
MIFS-C	Mutual Information Feature-Selection Type C
NB	Naive Bayes
NNet	Neural Network
SDI	Selective Dissemination of Information
SMO	Sequential Minimal Optimazation
SVC	Support Vector Classifier
SVMs	Support Vector Machines
TC	Text Categorization
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TS	Term Strength
WAvg	Weighted Averaging
Weka	Waikato Environment for Knowledge Analysis
WMax	Weighted Maximum
WSD	Word Sense Disambiguation

CHAPTER 1

INTRODUCTION

1.1 Background

The human experts usually do the documents cataloguing and indexing manually. With the growth of online information, and sudden expansion in the numerous electronic documents provided on the web and digital libraries, there is difficulty in categorizing in both electronic documents and traditional library materials using only a manual approach [1].

To solve these problems as well as improve the efficiency and effectiveness of document categorization at the library setting, so many researches have used automatic document classification methods to categorize library items, and machine learning researches which has advanced quickly in current years. [1], and this work is one of those researches. The main idea of text categorization is to allot textual documents data according to one or more predetermined topic codes on the basis of knowledge accumulated in the training process [2, 3].

The major part of text categorization is the feature selection, and many researches deal with various feature selection methods. Feature selection can be defined as selecting the best words / tokens of a text document that can help in categorizing of that document [3].

The procedure of text categorization is alloted a set of document, D, and some pre-determined set of categories, C, then categorize documents, D, according to appropriate categories, C, as best as possible.

Text-categorization is a very good technique that uses labeled training data for learning the classification system, such as (BayesNet, NaiveBayes, Trees ...etc.) and then automatically classes the remaining text by applying the learned system. For instance, it can determined the words such as \GalaxyS5", \GalaxyS4\, or \Note5" those are related to category of technology named \Samsung, and then their documents must belong to that category [4].

Simply the feature selection determines the important words\tokens by using some measure that can keep informative words and cancel non-informative words. It is known, there is a difference between training an algorithm to keep informative words and to remove the non-informative [4]. For example, an adult person trained to check for instance of word \Galaxy S5 and put it in related document in \Samsung category, of course, is a different job, Feature selection metrics that are studied, implemented, are used to provide a good results for enhancing performance of learning machine on the documents collection.

1.2. Literature Survey

The field of the Text Categorization established and founded many important applications such as (Electronic Government, Information Retrieval, Electronic Libraries......etc). In practical, Text Categorization has proved to be suitable tool in retrieval of information field by a numerous researchers.The categorization system illustrates news of story by assigning the index of terms to news stories depending on their content by using knowledge-based techniques [5]. It was implemented indexing for a huge database of technical summaries [6].

This study presented a comparative between two learning algorithms (Bayesian classifier and decision tree learning) on text categorization, and it is found that both the algorithms result acceptable performance and permit differentiation between false positives (FP) and false negatives (FN) [7].

It used the batch-mode learning method as proposed work to support vector machine to classify Arabic texts [8].

Many machine learning methods were used for text categorization, such as, nearest neighbor classifiers, decision trees, neural networks, regression methods, etc. [9]. In recent years, many researches have explored the usage of Support Vector Machines (SVMs) for text categorization and gave a nice result [10, 11, 12].

In 1998, Thorsten Joachims presented one of the most important studies related to SVMs for text categorization [13].

In 1999, Yang and Liu present a controlled study with statistical significant tests on the following machines learning algorithms (SVM, Naïve Bayes (NB), Neural Network (NNet), Least-Square Fit (LLSF) and K-Nearest Neighbor (K-NN) mapping) [14].

The critical matter in TC is reducing dimensionality of feature, and for improving the efficiency and accuracy of the classifiers used a feature selection, and it considers as one of the effective methods. In the literature, different feature selection methods have been presented [15].

The reducing of dimensionality is important issue in text categorization. Feature selection method improves the efficiency and accuracy of the classifiers by selecting only more discriminative terms in a dataset as features. In the literature, many feature selection methods have been presented and analyzed. [16].

In 1997, Petersen and Yang made a comparative study among five feature selection metrics (document frequency (DF), mutual information (MI), x2-test (CHI),information gain (IG) and term strength (TS)) on the Reuters and OHUMSED datasets by using k-NN and LLSF classification algorithms and in the case of global policy. They found that CHI and IG are the most successful [17].

In 2000, Yiming Yang had made a study on a comparative evaluation of a huge range of methods of TC, which included formerly printed outcomes on the Reuters corpus and new added experiments as well, in that he had got a global observation, kNN and neural for Naive Bayes approach[18].

In 2003, Sebastiani and

Debole, had got an excellent results by their propose work supervised term weighting (STW) scheme where they used three feature selection scores (IG, CHI and gain ratio (GR)) with tf-idf weighting on Reuters dataset with SVM in both local and global policies [19].

In 2005, Özgür et al made a comparison between Boolean weighting and tf-idf weighting on Reuters dataset with SVM again in both local and global policies, after that they discovered that tf-idf weighting performs better than Boolean weighting [20].

In 2007,Sherine N.and Yasser S. proposed an algorithm, where they focused on the removing of the redundant features and chose the maximal relevance to the categories, by depending on the mutual information measure, that algorithm was applied on the Reuters-21758 data set. After that, special features were inputted to the Naive Bayes, and Support Vector Machines [21], the results were found better than MIFS-C and information gain measure algorithms presented in [22].

In 2009, Feng X., Tian J. and Liu Z. proposed TCBLDF method for text categorization, and the experimental results showed that the proposed method is more effectual for TC than Naive Bayes and an SVM learner with linear kernel function [23].

In 2011, Zhilong Z., Haijuan W., L. H. and Zhan proposed filter based on feature selection, named categorical document frequency, and the empirical results were more benefit and more reliable to processing large-scale text data, so the computation cost of document frequency is lower than information gain and chi-square [24].

In 2013, Hong Zhang, Yong R. and Xue Y. studied on the IG algorithm and improved it, the enhancement of (IG) which was necessary because the distribution of data set was imbalanced and which may have a bad effect on classification.The comparative experiments show that improvement has given an excellent progress in performance [25].

2014, I. Budiselić, G. Delač and K. Vladimir studied the StackExchange question classification problem and they gave a solution to it. The goal of their studies was to get an accurate and an efficient text classifier for relatively simple problem domains in only a few hours and their discussion was focused on feature selection and example representation [26].

1.3. Motivation and Objectives

Our basic stimulus in this study is to obtain the most affluent features for the Text Categorization problem as well as use them maximum in order to gain from them in the most effective way. Preliminary experiments are executed using all the probable features without using a selection procedure. For the advanced experiments, the main feature selected algorithm (pruning) is executed on the basis of term's frequency in the whole dataset. Performance of the pruning levels with different feature types (word, dependency) and different datasets are boosted. In the earlier tests, we used (weight= tf-idf) as well as a substitute method for the selection of the feature, which is actually one of the most popularly used feature of selection metrics. We got similar success rates as the pruning application when only the words were used, but the success lowered when the dependencies were counted in the feature vector. Thus, it was decided to go on with the pruning technique. As by using a feature selection metric on the dependencies may lead to a thorough analysis and we leave the study of combining feature selection metrics in text categorization

It was necessary here to implement many steps to satisfy the objectives of Text Categorization (TC) for Reuters-21578 corpus and make that TC easy as possible.

Here firstly, Reuters-21578 has been used and this collection comprises of 21,578 documents, and which was chosen from Reuter's newswire stories, the documents of it were split into training and test sets. Every document had five category tags, namely, TOPICS, PLACES, PEOPLE, ORGS and EXCHANGES. Each category had number of topics that were used for a document, but this study focuses on the TOPIC category only.

Secondly, Pre-processing of Text was done by using three approaches. Stemming approach to separate text into individual words. Stop word removal is used for to remove common words that usually are not necessary for text classification, for example removing words such as (she, he, it, we, is, are, the, this, that, etc.). The stemming is used for to normalize words derived from the same root for example, for instance computer to compute, and studding to study, so on.

Thirdly, Feature Extraction is meant to use each word as a feature after the apply term frequency (TF) as feature value, and use TF*IDF (inverse document frequency) as feature value.

Finally, the weight term refers to (TF*IDF*Log (total-number of documents / number of documents containing term)), and it represented a feature value.

The goal of this study is to build a framework for the TC problem based on some learning machine classifiers such as **(**BayesNet, NaiveBayes, Trees**)** with document frequency measurement for feature reduction that is a difficult problem when documents have a huge of keywords. All these field expressions are done by using c ++ programming and Weka 3.7 Model.

1.4. Outlines of the Thesis

This thesis contains six chapters. All the necessary information about the Text Categorization, feature selection and representation and document processing…etc is given there and a brief outline of the thesis is given below:

Chapter 1 is an introduction to text categorization approach, literature Survey motivation and objectives of this work and finally outlines of this thesis.

Chapter 2 includes a brief review of text categorization is given. First, text categorization definition is reviewed. Why is it needed for using a Text Categorization,is the question asked here. A variety of applications of text categorization are described, and advantages of text categorization are explained. In addition, tasks of text classification are introduced.

Chapter 3, text document pre-processing is given. Document representation (DR), bag of words (BOW), tokenizing and parsing the documents, stop word removal, stemming and term weighting are processed here to determine the most appropriate features words for classification.

Chapter 4 introduces a definition of feature selection and reasons of using the feature selection are explained as well, Document frequency (DF), Mutual information- explanation (MI), Chi-square statistics (CHI) and GSS (Galavotti-Sebastiani-Simi).

Chapter 5 introduces a solution of the problem, Reuters-21578 dataset**,** explains the System Structure Processes, The main steps of the system process, Definition of weka (Waikato Environment for Knowledge Analysis), the features of Weka are explained, summarization of evaluation measurements formulas are expained and the evalution measures definitions. Finally**,** definition of learning machines algorithms.

Chapter 6 includes the Results part

Chapter 7 includes the conclusion and future work

CHAPTER 2

A BRIEF REVIEW OF TEXT CATEGORIZATION

2.1 Text Categorization Definition

Text Categorization can be described as a supervised learning function of automatically assigning a document based on the likelihood derived from a set of labeled training into a set of predefined categories [27], and this is a task of increasing importance in natural language processing and information retrieval (IR) systems [7].

Text categorization is defined as the issue of automatically allocating predetermined classifications to many text documents which are available free. When more and much more text documents are available on the web sites, the indexing and summarization of document content make the information retrieval more easily done. In recent years, there are many quantity of statistical classification procedures and machine learning techniques have been implemented to text categorization[23].

Text categorization is the procedure of assigning a given natural language texts to predetermined classifications on their content bases. [23].

Text categorization (TC) is also recognized as topic observatory, or text classification is the function of automatically separating a set of documents according to their groups, or types from a predetermined set [28]. The text categorization approach is shown in Fig.1.

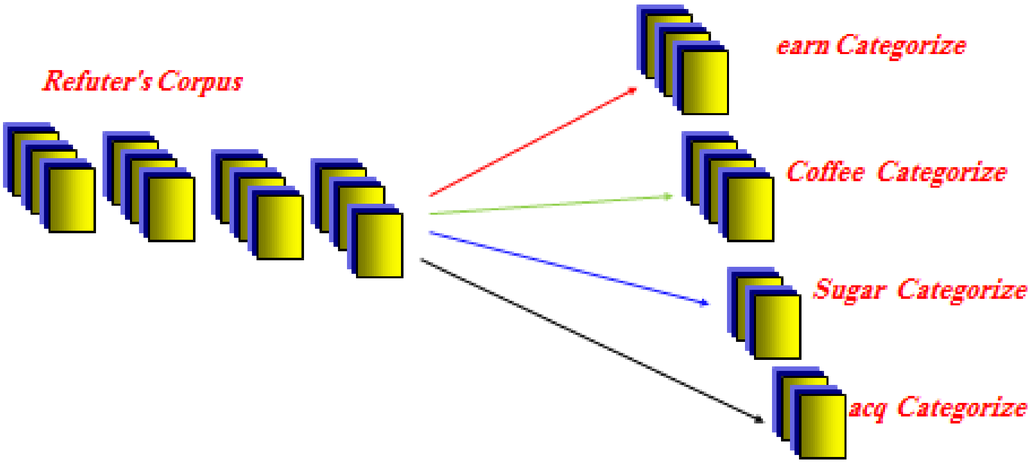

Figure 1 Text Categorization Approach.

2.2 Why is it needed for Using a Text Categorization?

It needed for text classification automatically, because the classification of thousands of text document manually, is more expensive and much more time consuming task. Therefore, constructed Automatic Text Classification is a necessary solution to get a better accuracy, efficiency and it requires less consumtion of time than manual text classification [29].

Manual assignment of categories to documents is usually used in text categorization, but this way is not suitable for a huge number of documents, because it needs a huge human labor, and it is impossible to categorize all the documents. Therefore, many automated text categorization have been proposed [30] using Text Categorization approach it gets much faster and cheaper classification than human labor [31].

2.3 A Variety of Applications of Text Categorization

TC is an important approach in information retrieval, Electronic Government, Electronic libraries….etc.

- A major application of text categorization systems is to examine the documents and assign subjects categories to those documents, and this duty is significant to back information retrieval, or to help human indexers for assigning such classifications. [32].
- Text classification may be ustilized to select documents or sections of documents that are unlikely to contain without increasing the cost of natural language processing [33].
- A huge use of text categorization for extraction data in natural language processing systems [34].
- Text categorization components can help to guide texts to category-specific processing mechanisms [35].
- Text categorization systems is same of human categorization judgments, but is automatically done; TC is another way to produce human categorization judgments by using inductive learning machines to select categories to documents based on the word is contained within those documents. Such an approach can help effort made by human in constructing a text categorization system, and that system or helping human indexer is led to produce a huge database of documents.

2.4 Advantages of Text Categorization

There are many benefits of text categorization in different fields of text documents, news, stories, messages, electronic articles and books, electronic trade,..etc. especially used via web.

- Text Categorization helped to emerge a new science, such as Authorship attribution: it is defined as the science of determining the author of a text document, from a predefined set of candidate authors or deducing the characteristic of the author from the characteristics of documents written by him [36].

- Text categorization classifies the documents according to the sender and message type, so that it allows an efficient distribution of documents via fax or email with less implementation time compared to manual processing of mailing or faxing, that consumes more time [36].

- Automated survey code, it has several advantages. For example, in the social sciences, it is begun from the simple classification from respondents on the basis of their answers based on the extraction of statistics on customer opinions, political, and health satisfaction etc. [36].
- Word sense disambiguation (WSD), sometimes, gives the occurrence in a text of an ambiguous term.

For example, text has vocabulary words which have different meaning, such as; the word bank which may have (at least) two contrasting senses in English, such as the Bank of England (a financial institution) or the bank of river [36].

- Text Filtering, TF is meant to filter text document that contains specific or several keywords, for instance, e-mail filters and newsfeed filters [36].
- Finally, Document clustering, is meant to collect the similar documents into clusters, without which we cannot have any external sources which has information on the correct clustering for documents.

2.5 Tasks of Text Classification

Classification is considered as a vague term in information retrieval, but it usually refers to procedures of catagorizing of entities. Text Classification is a suitable term as it actually collects numerous tasks of information retrieval and text categorization. We have explained some of tasks in this section; we have also focused on the text retrieval and text categorization as well[37].

2.5.1 Text retrieval

Text retrieval is the system of selection of a subgroup of a database according to user's request. The overview of text retrieval system is a complex system, which sorts documents into two classes, one of them is displayed to user and other is not.

There are many advanced text retrieval systems which do not select documents those which have an importance only, but also computes the degree of membership of documents in a class. In the following the text retrieval process [37].

- **Indexing:**

It must convert the raw documents into an expression, these expression are called documents representation, and these must be matched with text retrieval software.

- **Query formulation:**

Information retrieval software translates user's request, and that request is sometimes in a form very similar to that used by the system, such as a Boolean expression over words, but in other cases the connection may be less direct in case of using natural language question or example document by user, so IR software chooses important words only, and considers them as feature in the classifier. It actually uses the term query to refer to form of user request compared to documents.

- **Comparison:**

IR system compares the user query to the stored documents either implicitly or explicitly and produces the classification decision of which document will retrieve them.

- **Feedback:**

  The initial retrieval results of query rarely match exactly to the documents desired by user. Many iterations of modifying the query are often necessary to give an acceptance results.

2.5.2 Text categorization

It is the classification of documents according to a set of one or more pre-defined categories. The steps of text categorization can be considered as of the same steps of text retrieval, though some details are significantly different [37]:

1. **Indexing:**

It is the same as used in text retrieval. The speed of indexing is often more critical than in text retrieval, since there are many numbers of documents which may need to be proposed in real time.

2. Categorization Formulation

It is similar to text retrieval, it requires specification of ways to take a decision to which category a document should be signed, dependent on its text representation structures. It plays the same role as the query in a text retrieval system does. While text retrieval queries are typically temporary structures [37].

3. Comparison

In text categorization system, it requires a binary decision from each category for each document. There is a difference between text categorization system and text retrieval system, because one document may correlate to numerous classifications at once, and the suitable decision made may depend on relationships among those classifications.

4. Adaptation

It plays two roles in text categorization systems; both are different from its roles in text retrieval.

- There are large numbers of manually categorized document already available, when a categorization system is constructed.
- The users can communicate their answers to the categorization system maintainers, who may customize, add, or remove categorizers.

2.5.3 Text routing

It is defined a selective dissemination of information (SDI), and text routing joins forms of text retrieval as well as text categorization [37].

2.5.4 Term categorization

It is like text categorization, in that bit of text responsible to predetermine categories [37].

Figure 2 shows Structure of text retrieval system.

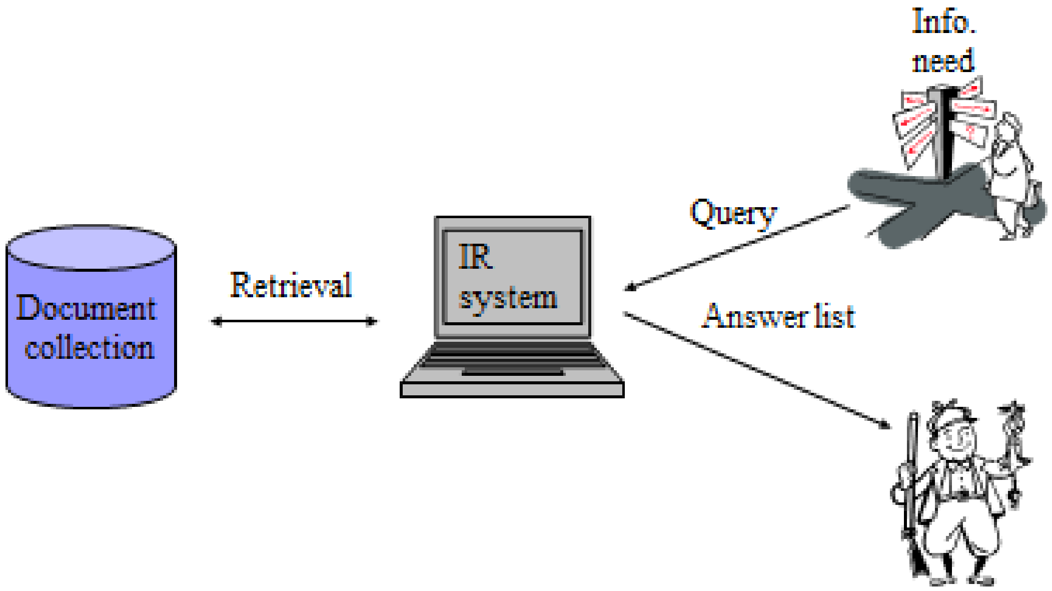

Figure 2 Structure of Text Retrieval System.

CHAPTER 3

TEXT DOCUMENT PRE-PROCESSING

3.1.Document Representation (DR)

The requirements of applying machine learning techniques are document representation, DR is the process of converts the unstructured text into a structured data as a vector in order to classify the text documents, after that applies machine learning techniques [38]. Document is illustrated as a vector (D)and each dimension in that vector corresponds to term in the term space of the document collection [16]. Illustrating the documents as vectors is shown in Fig.3.

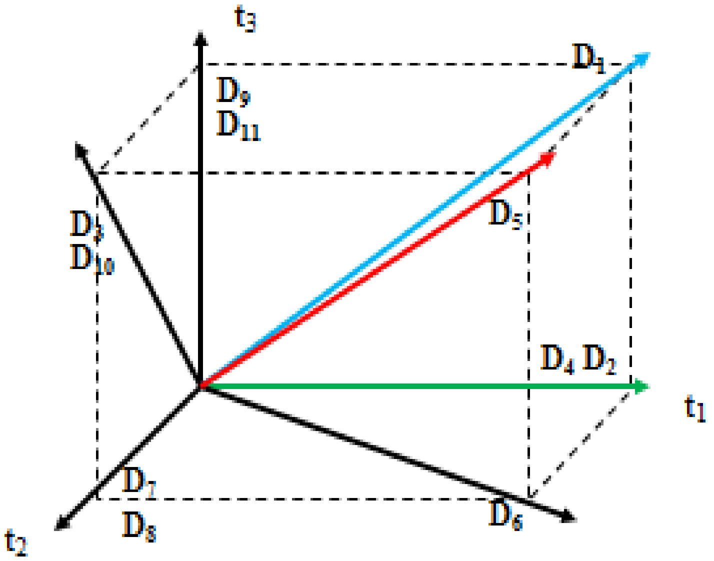

Figure 3 Illustrating the Documents as Vectors.

3.2. Bag of Words (BOW)

In this study, Bag of Words (BOW) is used in vector space model, and each term represents as a distinct single word. BOW can be created, which can be used to categorize test documents.From the BOW, we can keep informative words, and remove non-informative words, and this idea of choosing informative words, and removing the rest is called Feature Selection.

Although Bag of Words (BOW) is considered a simple method, but high dimensionality in the feature space is an important issue, and it is necessary to reduce the dimensionality, so it is applied to some preprocessing metrics which are described by the following tasks:

3.3. Tokenizing and Parsing the Documents

In the first step, non-alphabetic characters such as numerals, special characters and date and all the HTML mark-up tags are deleted from the documents in the dataset by use the parsing of the documents approach. Tokenization is used to separate text into individual words by words splitter tool.

For example: " We're attending a tutorial now."

It becomes as : we're attending a tutorial now.

3.4. Stop Word Removal

It is to remove common words that are usually not useful for text categorization, and the overly common words, such as prepositions, pronouns, and conjunctions in English like "it", "he", "she", "is", "are", "am", "in", etc. This occurs so frequently that they cannot give any useful information about the content and be discriminatory for a specific class. These words are so called “stop words”. We use the stop word list built by Salton and Buckley for the SMART system at Cornell University to eliminate common words. The list consists of 571 words given in Appendix A [39].

3.5. Stemming

Removing stop words is a necessary step and causes an efficient reduction in the dimensionality of the feature space to a reasonable number, but this is not enough. It needs to do a stemming approach.The stemming is a preprocessing for discovering the root morphemes of the words, and the goal of it is to normalize words derived from same root, for example (teaching = teach, attended = attend etc.) we use Porter's Stemmer that is the most widely used algorithm for word stemming in English. Porter's Stemming Algorithm is a process for deleting the commoner morphological as well as inflectional affixes from words [40, 41].

In other words, Porter"s Stemming Algorithm is based on only morphological issues. For example, the words “computer”, “computers”, “computing” and “computers” are stemmed from the same root “comput”. After stemming, terms that left a single character are also removed since they cannot give any information about the content of a document.

3.6. Term Weighting

As already it is mentioned in section 3.1., each document is represented as a vector D.

$$D = (w_1, w_2, \ldots\ldots\ldots w_n) \qquad (3.1)$$

Where, w_j is the weight of term j of document D. There are many methods proposed to compute these term weights [42]. However,there are three major assumptions that are valid for all computations [43].

1. The document that has more term is not more important than document that has less terms.
2. The term has multiple appearances in a document, that is not less important than term has a single appearances in a document.

3. Rare terms are no less significant than frequent ones.

The term frequency-inverse document frequency (tf-idf) weighting is an important method in computing of weight of the term of document, and it is one of the widely used weighting methods that take into account these properties.

1. Length-normalization meets the third assumption.
2. (tf) formula meets the second assumption.
3. (df) formula meets the first assumption.

Thus it is applied *tf-idf weighting* method in this study wh formula is given below:

$$wij = tf_{ij} . \log(N / df_{ij}) \qquad (3.2)$$

Where:-

w_{ij} : is the weight of a term *i* in document j.

tf_{ij} : denotes the frequency of the term *i* in document j.

df_{ij} : denotes *the number of documents* in which a term *i* occurs in the whole documents.

N : is the total number of documents, and it is (11414) documents in this study.

The $tf - idf$ weighting considers that if a term occurs more often in a document, it is more discriminative whereas if it appears in most of the documents, then it is less discriminative for the content.

Terms and their weights can be arranged in table, which is called Term Weighting Table , and table1. is summarizes the relation between documents, terms and weights of those terms.

Documents **Terms**	**d1**	**d2**	**d3**	**...........**	**dM**
t_1	w_{11}	w_{12}		w_{2i}	w_{1N}
t_N	w_{N1}				w_{NM}

Table 1 Term Weighting Table.

In the next page, is arranged document's pre-processing in chart, and here is prepared an example for text representations for finding documents that are most similar to the query. The user searches about the weather today and the most similar documents to that question are D1 and D2. Where, D1 is the weather today which represents it was raining but the weather yesterday was sunny, and D2 is the weather today which shows raining and is similar to last year but the wind today was stronger.

The question by a query is; what is the weather today, today?

The pre-processing of text documents D1, D2, and Q are prepared according to the steps in previous sections. The following figures (Fig. 4. And Fig. 6.) explain the procedures of text documents pre-processing and the answer related to that question is declared by the query.

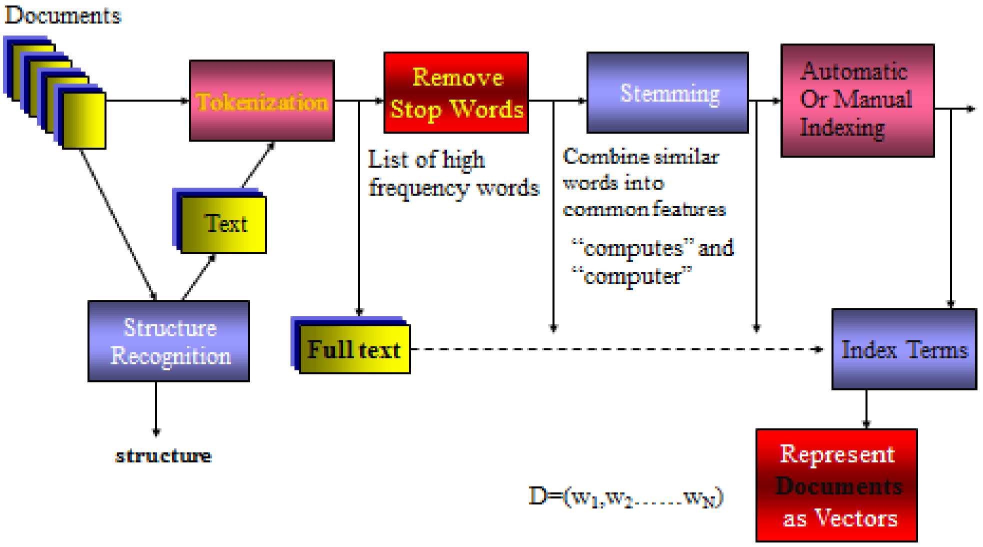
Documents
Tokenization
Remove Stop Words
Stemming
Automatic Or Manual Indexing
Text
List of high frequency words
Combine similar words into common features
"computes" and "computer"
Structure Recognition
Full text
Index Terms
structure
Represent Documents as Vectors
D=(w1,w2……wN)

Figure 4 The Document Preprocessing Chart.

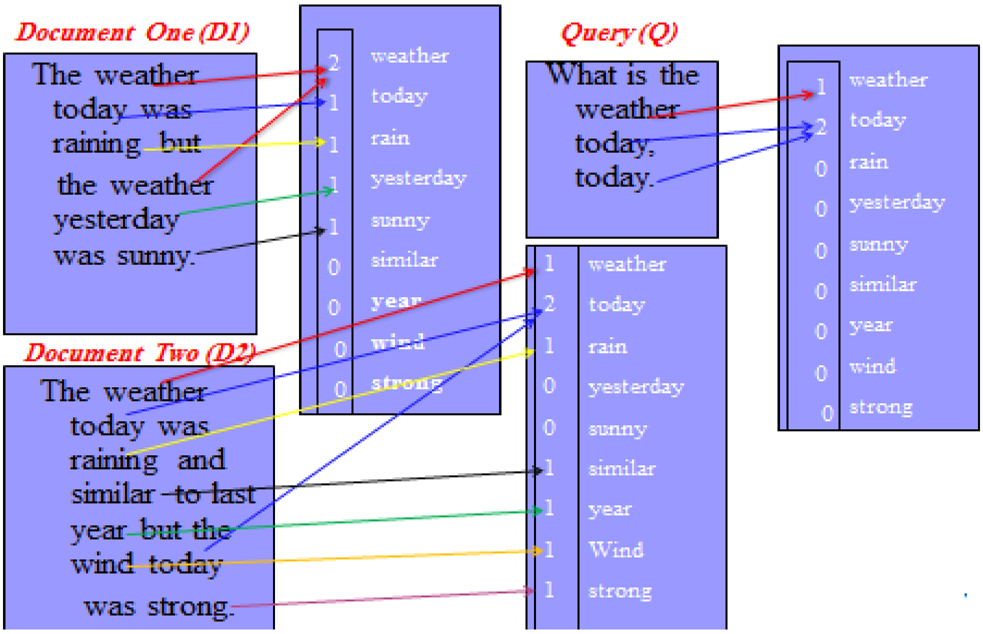

Figure 5 Text Document Representing.

Figure 5 shows that document D2 is most similar to the query, but document D1 is less similar to query.

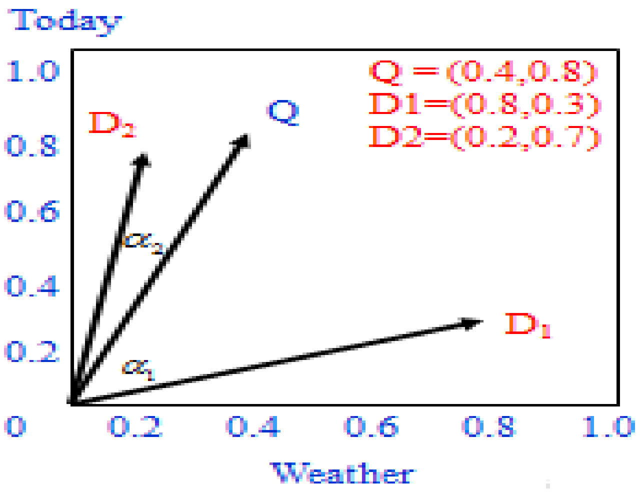

Figure 6 Vector Space Models for D1, D2 and Q.

CHAPTER 4

FEATURES SELECTION

4.1. Definition of Feature Selection

Feature selection is defined as a selection of a set of the features/tokens/words available to describe the data. [44]. or is a major step in text categorization approach, which gives more accuracy on the classification of documents. [45]. OR is a method for contraction of the dimensionality of the dataset by deleting irrelevant features for the classification. [46]

4.2. Reasons of Using the Feature Selection

Many researches have done work on the importance of feature selection in text categorization, and it should be used when classified data is according to classes [47]. The main issue of text categorization is the high dimensions of the features space. For many learning algorithms, those high dimensions are not allowed. Furthermore greater number of these dimensions are not related to text categorization; even some noise data reduce the correctness of the classifier of the learning machine. [48]

It is not suitable to use all the features collected from training documents in text- categorization method, it should reduce the number of features used for impersonation of documents, and that is very necessary for using most of the machine learning algorithms. [46]

For the above mentioned reasons, many techniques are used to reduce the amount of features / tokens/words, so that new documents can be categorized easily, quickly as well as small amount of computations.

4.3. Benefits of Feature Selection

Below are given many benefits for using feature selection

1. Facilitation or acceleration of the computations with a little loss in quality of classification [44].
2. Elimination of non-informative and noisy features and reduction of the feature space to a manageable size [49].
3. Decrease the dimensions of feature space and enhance the gain, performance and precision of the classifier of learning machine [46, 48, and 50].
4. Enhance classification computations, efficiency, effectiveness, and accuracy [50, 51].
5. Helps keep computational requirements of text-categorization algorithms that do not scale with the feature set size and dataset size [46].

4.4. The Global and Local Dictionaries

There are two main dictionaries to apply feature selection in text categorization: local and global dictionaries (local and global policies).

1. In the local dictionary, a different set of features is selected from each category.
2. In the second dictionary, a single set of features is selected from all categories.

Several studies have been done to use local and global dictionaries (local and global policies).

In local dictionary, a contrasting set of features is chosen from each independent of the other categories, and that dictionary happens to maximize the classification process for each category by choosing the most important characters in that category.

In 1994, Apt´e, Damerau & Weiss used local dictionary, and it was for the most relevant features in each category, they selected from each category the words that were similar to the category dictionary.

Use of local dictionaries suffers from being a domain as well as for language-independent approach. In 1994, the top IG features from each category were selected by (Lewis & Ringuette). In 1997, Nget al. applied local strategy to three features scoring methods, namely Document Frequency (DF), Correlation Coefficient(CC), and X^2-Test.

The infrequent classes are penalized by global dictionary, when the classes were arranged in skew dataset from high to low by depending on selecting the most important features for the entire dataset [19].

Global dictionary is an alternative to the local dictionary, the goal of it to provide a global prospective of the training set by obtaining a global score from local feature scores. Threshold is then used to these global scores, where characteristics with the highest global score are maintained.

In 1997, Yang and Pedersen presented several ways to obtain global score from the local score:

1. Maximization.

2. Averaging.

3. Weighted averaging.

The weighted averaging techniques are common globalization techniques [52].

Yang and Pedersen used Maximization (Max) and Weighted Averaging (WAvg) for obtaining global scores from X^2-and MI.

In addition, they averaging (Avg) for DF and IG. Calvo and Ceccatto put forward the usage of Weighted Maximum (WMax), where features are considered by the category probability (Calvo & Ceccatto 2000). Equations 1, 2, 3, and 4 deliver the mathematical definitions of Avg, Max, WAvg, and WMax accordingly.

$$F_{Avg}(w_k) = \frac{\sum_{i=1}^{M} f(w_k, c_i)}{M} \tag{4-1}$$

$$F_{Max}(w_k) = \max_{i=1}^{M} \{ f(w_k, c_i) \} \tag{4-2}$$

$$Fw_{Avg}(w_k) = \frac{\sum_{i=1}^{M} p(c_i) f(w_k, c_i)}{M} \tag{4-3}$$

$$F_{Max}(w_k) = \max_{i=1}^{M} \{ p(c_i) f(w_k, c_i) \} \tag{4-4}$$

Where:

$f(w_k, c_i)$ is the score of the word w_k w.r.t. the category c_i, and *M* is number of classes in the training set.

4.5. Feature Selection Metrics

In this section, the five feature chosen metrics are studied, and the document frequency metric (DF) is used, since it is a good and the more commonly used one. Term frequency-inverse document frequency *(tf-idf)*, Mutual Information – Explanation (MI), chi-square statistics (CHI), Accuracy 2 (Acc2) and Galavotti-Sebastiani-Simi (GSS) are explained. The following section describes these metrics appearing in the literature.

4.5.1. Term frequency-inverse document frequency *(tf-idf)*

We select the standard *(tf-idf)* metric for term weighting in our procedures.The idea of the *(tf-idf)* feature selection is to selects the words with the highest *(tf-idf)* scores. This method gives the highest scores to the terms that appears in a few documents with a high frequency. In other words, if a term happens more often in a document, this means it is more discriminative whereas when it appears in most of the documents, then it is less discriminative for the content.

$$tf - idf = \log(N / df_{ij}) \qquad (4.5)$$

df_{ij}: symbolizes *the number of documents* in this a term *i* occurs in the whole documents.

N : is the total number of documents, and it is (11414) documents in this study.

4.5.2. Document frequency (DF)

The document frequency is considered as a good and common method, it measures the number of documents in which the term appears without class labels [53]. The purpose of this method is to eliminate the rare words which are assumed non-informative and misleading for classification. Document frequency feature selection method select the terms with the scores. Its formula is:

$$\text{DF} = \text{A} \qquad (4.6)$$

A: is the amount of documents that have the term, t, and also belong to class(c).

Those above metrics are used in this work, and it gave a good results.

4.5.3. Mutual information- Explanation (MI)

Mutual information method assumes that the "term with higher category ratio is more effective for classification" [51].
Mutual information can be calculated as follows using our already calculated A, B, C, D values:

Where,

A is the number of documents that have the term, t, and also belong to category, c.
B is the number of documents that have the term, t, but do not belong to category, c.
C is the number of documents that do not possess the term, t, but belong to category, c.
D is the number of documents that do not possess the term, t, and do not belong to category, c.
N is the number of training documents [17].

$$MI = \log \frac{A * N}{(A + C) * (A + B)} \qquad (4.7)$$

4.5.4. Chi-square Statistics (CHI)

In experimental sciences, chi-square statistics is frequently used to estimate how the observation results differ from the expected results. In other words, it measures the independence of two random variables.

$$CHI = \sum_{ij} \frac{oberved_{ij} - Expected_{ij})^2}{Expected_{ij}} \tag{4.8}$$

Chi square estimates the absence of between a term, t, and the classification, c [17].

Chi square, x^2, can be calculated as follows, again, using our previously calculated A, B, C, D values:

$$x^2 = \frac{N(AD - CB)^2}{(A+C)(B+D)(A+B)(C+D)} \tag{4.9}$$

4.5.5. GSS (Galavotti-Sebastiani-Simi) coefficient – explanation

Galavotti-Sebastiani-Simi (GSS) propose a simplified x^2 statistic. They remove the $\sqrt{N}$ factor and the denominator completely. They describe the $\sqrt{N}$ factor as being unnecessary. They also remove the denominator, $\sqrt{(A+C)(B+D)(A+B)(C+D)},$ by giving the reason that the denominator gives high Correlation Coefficient score to rare words, and rare categories [54]. The GSS can be computed as follows

$$GSS = AD - CB \tag{4.10}$$

Now that we have understanding of these five feature selection metrics, we can move on to the implementation part. Next chapter describes how to combine the term frequency-inverse document frequency *(tf-idf)* and Document Frequency (DF) to get the best results.

$$B = N_{all} - A \tag{4.11}$$

$$C = N \;\; - A \tag{4.12}$$

$$D = N_{all} - N - B \tag{4.13}$$

CHAPTER 5

THE SYSTEM STRUCTURE

5.1. Introduction and Solution of the Problem

Here, we put forward an inclusive analysis of the lexical dependency and pruning concepts for the text classification issue.

The pruning process filters characteristics with low frequencies so that a small number but more informative features stay in the solution vector. We evaluate comprehensively the pruning levels for words, dependencies, and dependency combinations for Reuters-21578 Dataset. The main stimulus in this chapter is to utilize dependencies and pruning efficiently in text classification and to have more fortunate results using much smaller feature vector sizes.After that document frequency is used, it measures the number of documents in which the term appears without class labels.

Text classification procedure is generally distributed into two main steps. The first step is to get a training set which is comprised of category known document, using the training based on classification model; the next step is to utilize the model to divide unfamiliar class of document. Here figure. 7, it exhibits the structural framework of the text classification procedure. Firstly, preprocess the text, then text with vector space model to represent, as well as the feature selection; then create and train the classifier; in the end , use the classifier to categorize the new text.

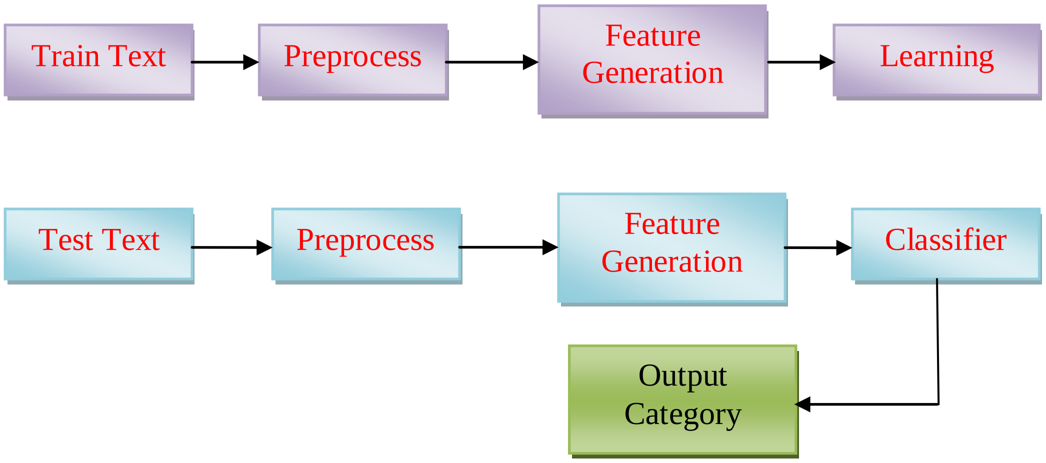

Figure 7 The structure framework of the text classification process.

5.2. Reuters-21578 Dataset

Here we utilize the corpus of Distribution 1.0 of Reuters-21578 text categorization test collection. This collection comprises 21,578 documents chosen from Reuter's newswire articles. The documents of this collection are distributed into training and test sets. Every document comprises of five category tags, that are, *EXCHANGES*, *ORGS*, *PEOPLE*, *PLACES*, and *TOPICS*. Every category comprises of a quantity of topics that are utilized for document assignment. We put a limit to our study to only *TOPICS* classification. To be more particular, we have employed the Modified Apte split of Reuters-21578 corpus that has 9,603 training documents, 3,299 test documents, and 8,676 unused documents.

5.3. System Structure Processes

In the next sections, text categorization system structure is presented; it consists of many steps. The functionality of each step is to characterized and then we describ interactions between individual steps.Figures (8, 9, 10, 11, and 12) demonstrate the system structure in detail.

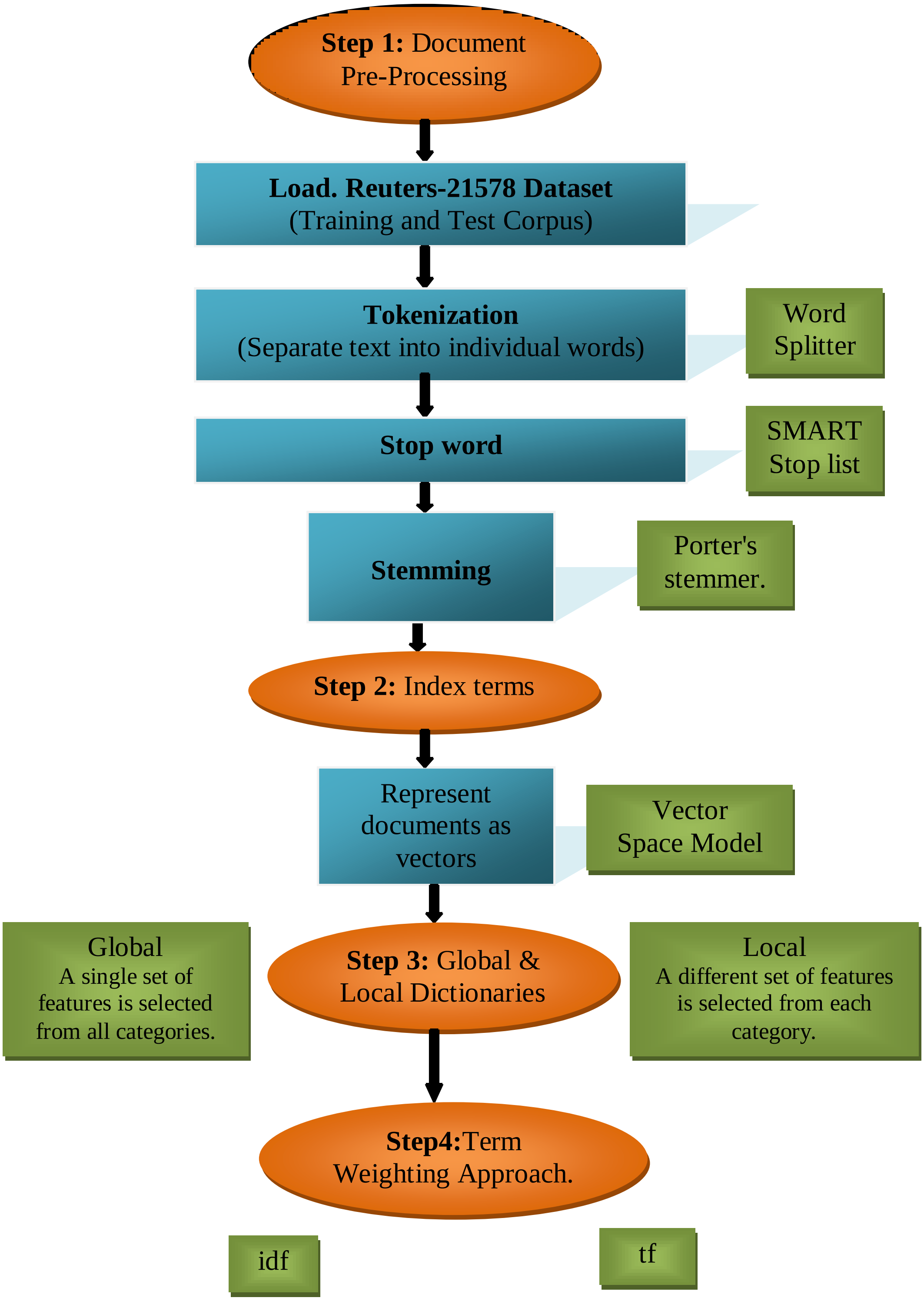
Step 1: Document Pre-Processing
Load. Reuters-21578 Dataset (Training and Test Corpus)
Tokenization (Separate text into individual words)
Word Splitter
Stop word
SMART Stop list
Stemming
Porter's stemmer.
Step 2: Index terms
Represent documents as vectors
Vector Space Model
Global A single set of features is selected from all categories.
Step 3: Global & Local Dictionaries
Local A different set of features is selected from each category.
Step4: Term Weighting Approach.
idf
tf

Figure 8 Preprocessing, Indexing of Terms, Dictionaries, and Term Weighting Steps.

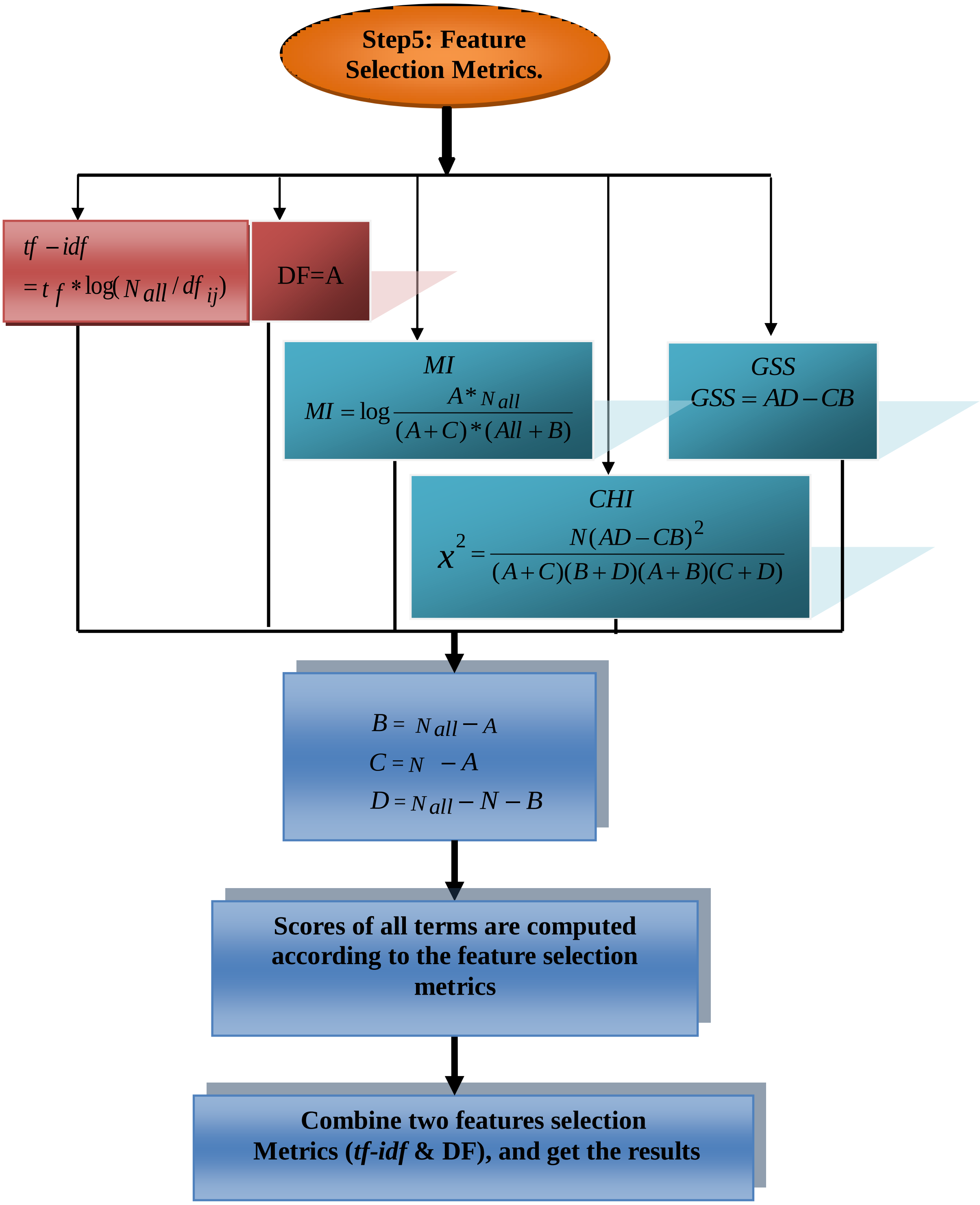
Step5: Feature Selection Metrics.
$tf - idf$
$= t_f * \log(N_{all} / df_{ij})$
DF=A
MI
$MI = \log \frac{A * N_{all}}{(A+C)*(All+B)}$
GSS
$GSS = AD - CB$
CHI
$x^2 = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)}$
$B = N_{all} - A$
$C = N - A$
$D = N_{all} - N - B$
Scores of all terms are computed according to the feature selection metrics
Combine two features selection Metrics (*tf-idf* & DF), and get the results

Figure 9 Feature Selection Metrics.

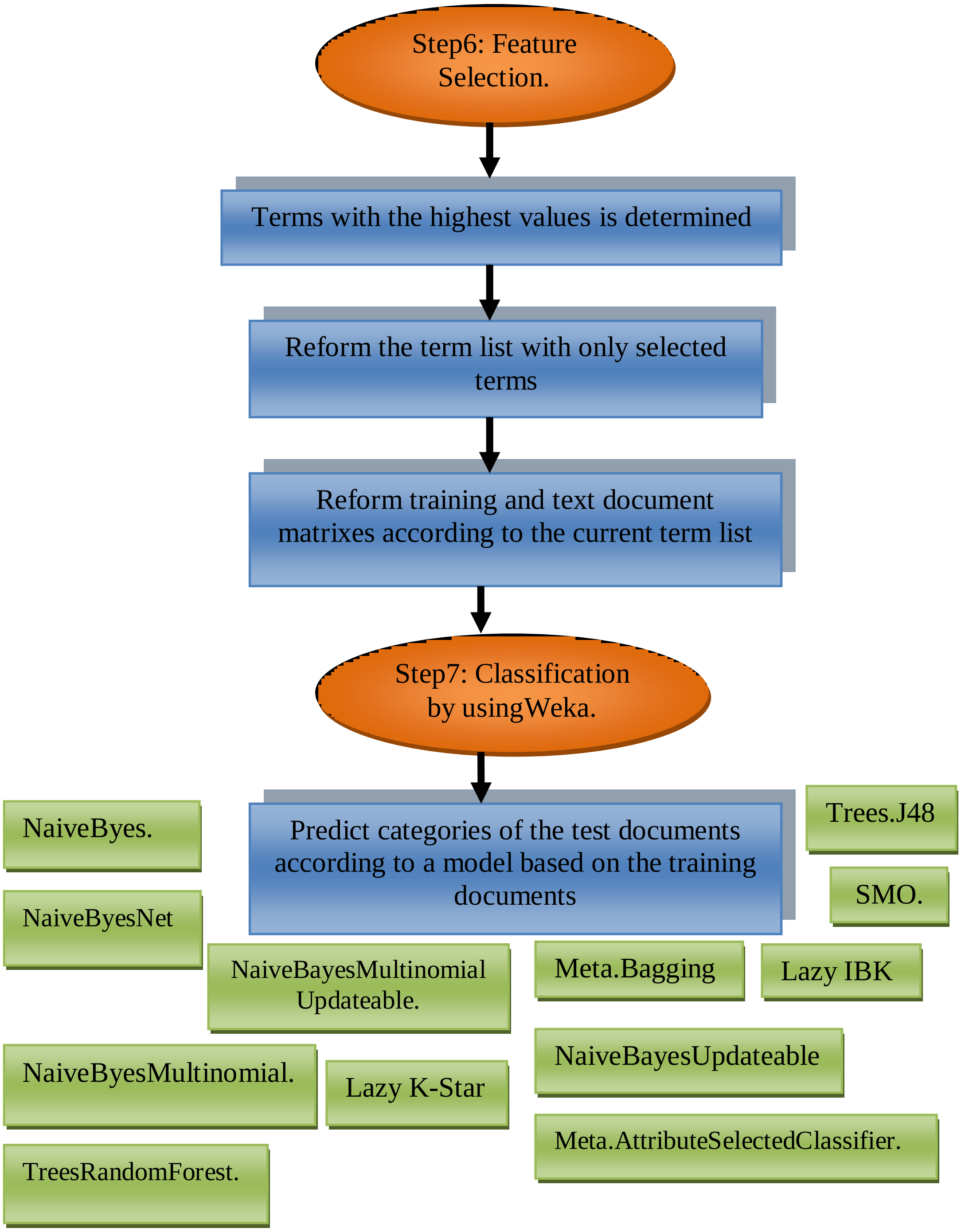
Step6: Feature Selection.
Terms with the highest values is determined
Reform the term list with only selected terms
Reform training and text document matrixes according to the current term list
Step7: Classification by usingWeka.
Predict categories of the test documents according to a model based on the training documents
NaiveByes.
NaiveByesNet
NaiveBayesMultinomial Updateable.
NaiveByesMultinomial.
Lazy K-Star
TreesRandomForest.
Trees.J48
SMO.
Meta.Bagging
Lazy IBK
NaiveBayesUpdateable
Meta.AttributeSelectedClassifier.

Figure 10 Feature Selections and Classification.

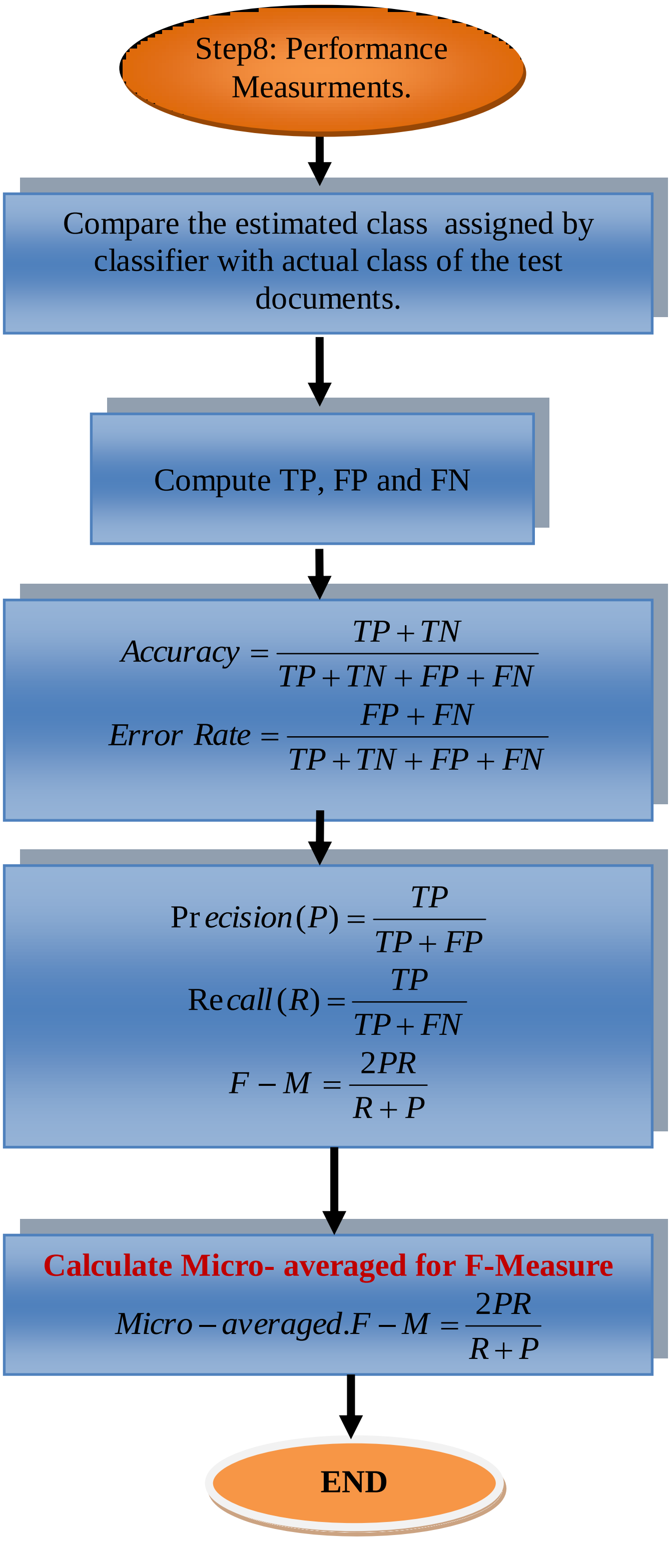
Step8: Performance Measurments.
Compare the estimated class assigned by classifier with actual class of the test documents.
Compute TP, FP and FN
$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}$
$Error\ Rate=\frac{FP+FN}{TP+TN+FP+FN}$
$\Pr ecision(P)=\frac{TP}{TP+FP}$
$\mathrm{Re}\,call(R)=\frac{TP}{TP+FN}$
$F-M=\frac{2PR}{R+P}$
Calculate Micro- averaged for F-Measure
$Micro-averaged.F-M=\frac{2PR}{R+P}$
END

Figure 11 Steps of System Measurements.

The main steps of the system are as follows:

5.3.1. Pre-processing of documents

First step is the pre-processing of the dataset (Reuters-21578 Dataset). In this step each document is parsed, non-alphabetic symbols and mark-up tags are removed by tokenization, the goal of this mission is to separate text into individual words by using Word Splitter tool, stop words are eliminated according to the stop word list of the SMART system and then each word is stemmed using Porter's stemmer. At the end of these processes the category list, term list, term matrix and category matrix of the training documents and term matrix and category matrix of the test documents are created.

5.3.2. Indexing of Terms

The terms are indexed using some types of method, the commonly used one and the the most successful text representation model is this:(Vector Space Model).

5.3.3. Global and local dictionaries process

There are two types of dictionaries are applied in text categorization: local and global dictionaries.

1. In the local dictionary, a different set of features is selected from each category.
2. In the global dictionary, a single set of features is selected from all categories.

5.3.4. Term weighting approach

- The tf (term frequency) as feature value, and is calculated for each term according to below formula.

$$tf_{ij} = (0.5 + \frac{0.5\, freq_{i,j}}{\max_{i,j} freq_{i,j}}) \qquad (5.1)$$

Where

$freq_{i,q}$ = Numbers of appears the word (i) in document (j).

The tf_{ij} weighting is calculated for each remaining word in the documents, or in another form use W_{ij} (weight of term) as feature value.

$$wij = tf_{ij}.\log(N / idf_{ij}) \qquad (5.2)$$

Where:-

wij : is the weight of a term (*i)* in document (j).

tf_{ij} : denotes the frequency of the term (*i)* in document (j).

df_{ij} : denotes *the number of documents* in which a term (i*)* occurs in the whole documents.

N : is the total number of documents.

5.3.5. Feature selection metrics process

There are five feature selection metrics(*Tf-Idf*, DF, CHI, Acc2, and GSS) which are studied here, and document frequency (DF) is used in this study as well.

5.3.6. Feature selection

The sixth step is feature selection that reduces the dimensionality by ranking all terms according to their importance estimated by combination and then selecting a given number of terms from the term list with the highest values. After selecting, the topic list, term list, and topic and term matrixes of the documents are reformed according to the selected features.

5.3.7. The classification by using Weka

In the seventh step, the category of the each test document is predicted according to the model derived from the training documents by using twelve algorithm learning machines in Weka (Waikato Environment for Knowledge Analysis) package.

5.3.7.1. Deffinition of weka(Waikato Environment for Knowledge Analysis)

Weka is open source software beneath the GNU General Public License. System is produced at the University of Waikato in New Zealand. “Weka” present the Waikato Environment for Knowledge Analysis, and indicate to a large flightless New Zealand rail with heavily constructed legs and feet.. The software is allowed to be used freely at http://www.cs.waikato.ac.nz/ml/weka. The system is written utilizing object oriented language Java.There are many different levels at which Weka can be made use of.

Weka gives implementations of state-of-the-art data mining and machine learning algorithms. Weka possesses modules for data preprocessing, classification, clustering and association rule extraction.

5.3.7.2. Primary features of weka

- 49 data preprocessing tools.
- 76 classification/regression algorithms.
- 8 clustering algorithms.
- 15 attribute/subset evaluators + 10 search algorithms for feature selection.
- 3 algorithms for finding association rules.
- 3 graphical user interface.

5.3.8. Performance Measurements of System

In the eighth step, the performance of the classifier is evaluated according to the F-measure results. In order to compare the predicted categories assigned by classifier with the actual categories of the test documents, first of all the quantity of True Positives, False Negatives and False Positives are determined, then precision and recall is computed using these values and finally the micro- and macro-averaged F-measures are calculated from precision and recall.

5.3.8. 1. The Evalution measures definitions

The following evalution measures are:

- **Accuracy: Shows** a percentage of documents properly classified by the system.
- **Error Rate Inverse of accuracy:** These are the percentage of documents wrongly classified by the system.
- **Precision**: This is percentage of relevant documents appropriately retrieved by the system (TP) with reference to all documents retrieved by the system (TP + FP), in other words: Precision is equal to retrieved relevant documents/ retrieved documents (i.e. how many of the retrieved books are relevant?).
- **Recall:** This shows the percentage of relevant documents properly retrieved by the system (TP) with respect to every documents relevant for the human (TP + FN), in other words, Recall equals to retrieved relevant documents / relevant documents.
  (i.e. How many of the relevant books have been retrieved?).
- **F-Measure:** This combines in a single measure Precision (P) and Recall (R) giving a global estimation of the performance of an IR system.
- **True Positive (TPi):** This is the number of documents that are assigned properly to class (i).
- **False Positive (FPi):** This is the quantity of documents which are assigned incorrectly to class (i) by the classifier but that in actual are not part of class (i).

- **True Negative (TNi):** is the number of documents not assigned to class (i) by the classifier but which actually do not belong to class (i).
- **False Negative (FNi):** is the number of documents not assigned to class i by the classifier but which actually belong to class (i).

Table 2 shows relation between relevant and non relevant approach, and retrieved and not retrieved terms.

	Relevant	**Not Relevant**	
Retrieved	True Positive (TP) = a	False Positive(FP)=b	a+b
Not Retrieved	False Negative(FN)= c	True Negative(TN)=d	c+d
	a+c	b+d	a+b+c+d=n

Table 2 Relation Between Relevant and Non-Relevant Approach**.**

5.3.8. 2. The summarization of evaluation measurements formulas

The relations between evalutioin measures are explained in following formulas, and according to table(2) in section (5.3.8.1.) are:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{a+d}{a+b+c+d} = \frac{a+d}{n} \qquad (5.4)$$

$$Error\ Rate = \frac{FP+FN}{TP+TN+FP+FN} = \frac{b+c}{a+b+c+d} = \frac{b+c}{n} \qquad (5.5)$$

The performance evaluation feature selection approaches is computed by used the F-Measure metric, and it is common metric which is equavilant to the harmonic mean of Recall (R) and Precision (P) [59].They are explained as given below :

$$\mathrm{Pr}\,ecision(P) = \frac{TP}{TP+FP} = \frac{a}{a+b} = \frac{\mathrm{Re}\,trieved\,\mathrm{Re}\,levent}{\mathrm{Re}\,trieved} \qquad (5.6)$$

$$\mathrm{Re}\,call(R) = \frac{TP}{TP+FN} = \frac{a}{a+c} = \frac{\mathrm{Re}\,trieved\,\mathrm{Re}\,levent}{\mathrm{Re}\,levent} \qquad (5.7)$$

$$\begin{aligned} F-M &= \frac{2PR}{R+P} \\ &= \frac{2(TP)}{FP+FN+2(TP)} \\ &= \frac{2(\mathrm{Re}\,trieved\,\mathrm{Re}\,levent)}{\mathrm{Re}\,levent+\mathrm{Re}\,trieved+2(\mathrm{Re}\,trieved\,\mathrm{Re}\,levent)} \end{aligned} \qquad (5.8)$$

Figure. 12 explains the idea behind the F-measure, where red circle(right circle) represents all the defective set and the blue circle(left circle) represents the set that were classified as defective by a classifier. The intersection between these sets represents the true positive (TP) while the remaining parts represent the false negative (.FN) and the false positive (FP), and figure.13, it explains F-measure in another meaning.

Where red circle (right circle) represents retrievers set and blue circle (left circle) represents the relevant set that are catagorized by a classifier. The intersection between these sets represents the retrieved relevant set.

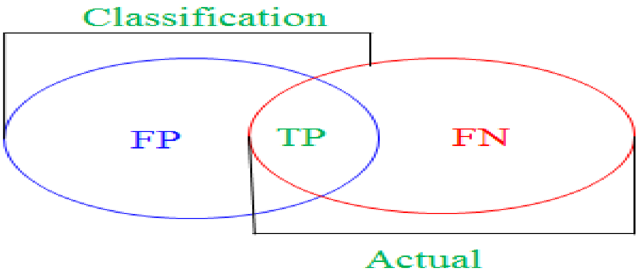

Figure 12 Demonstration of the F-measure According to TP, FP and FN Terms.

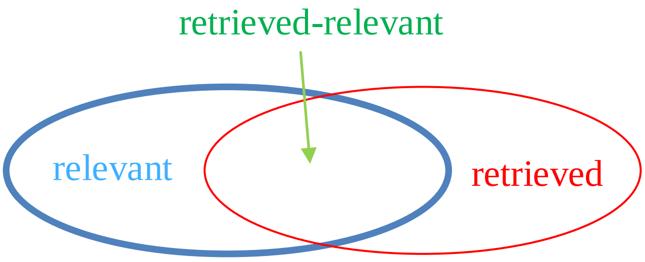

Figure 13 Demonstration of the F-measure According to Retrieved and Relevant Terms.

F-measure and micro-averaged F-measure which complete F-measure score of the whole classification problem which can be computed by utilizing these various types of averaging methods. [59]. Micro-averaged F-measure gives equal weight to each document and therefore it tends to be dominated by the classifier‟s performance on common categories while reflecting the overall accuracy better. Precision and recall are obtained by summing over all individual decision:

$$\Pr ecision(P) = \frac{TP}{TP + FP} = \frac{a}{a + b} = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c} TP + FP_i} \tag{5.9}$$

$$\operatorname{Re} call(R) = \frac{TP}{TP + FN} = \frac{a}{a + c} = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c} TP + FN_i} \tag{5.10}$$

Where C indicates the number of categories.

$$F - M = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{5.11}$$

$$Micro - averaged..F - M = \frac{2PR}{R + P} = \frac{2(TP)}{FP + FN + 2(TP)} \tag{5.12}$$

On the other hand Macro-averaged F-measure gives equal weight to each category regardless of its frequency and thus it is influenced more by the classifier‟s performance on rare categories. Precision and recall are firstly computed locally for each category and then F-measure is computed globally by averaging over the decisions of all categories:

$$\Pr ecision(P_i) = \frac{TP_i}{TP_i + FP_i} \tag{5.13}$$

$$\operatorname{Re} call(R_i) = \frac{TP_i}{TP_i + FN_i} \tag{5.14}$$

$$F - M = \frac{2P_i R_i}{P_i + R_i} \tag{5.15}$$

$$Macro - averaged.F - measure = \frac{\sum_{t=1}^{M} F_i}{M} \tag{5.16}$$

5.4. The learning Machines Algorithms

In this study, and by using Weka system, it was used the following learning machines algorithms:

5.4.1. Naive byes multinomial

The *Naive bayes* classifier the simplest of those models, in that it assumes that all attributes of the examples are independent of each other given the context of the class. This is the so-called Naive Bayes assumption." While this assumption is clearly false, in most real-world tasks, naive Bayes mostly performs catagorization very well. *Nominal* scales were mostly called qualitative scales, and measurements comprised of qualitative scales which were called qualitative data.But, the rise of qualitative research has made this usage confusing. *A multinomial model,* that is, a uni-gram language model with integer word counts (*e.g.* Lewis and Gale 1994; Mitchell 1997).

The *Naive byes multinomial* often performs even better at larger vocabulary sizes giving a small error[54].

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes.

5.4.2. Sequential minimal optimization (SMO)

This implements for training a support vector classifier (SVC). This implementation sunstitutes all missing values and changes nominal attributes into binary ones. It moreover normalizes all attributes by default. (In that case the coefficients in the output are based on the normalized data), not the actual data, this is essential for interpreting the classifier.

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Missing values, Numeric attributes.

5.4.3. Trees random forest

Random Forests are considered for catagorization of multisource remote sensing as well as geographic data. Numerous ensemble classification methods have been suggested in recent years. These methods have been proven to make better the classification accuracy considerably. This method is not sensitive to noise or overtraining, as the resampling is not based on weighting. Moreover, it is computationally much lighter than methods based on boosting and is somewhat lighter than simple bagging[57].

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large [58].

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, Empty nominal attributes, Date attributes, Missing values, Numeric attributes.
-

5.4.4. Meta bagging

It is the category for bagging a classifier to decrease variance, and can do catagorization and regression which depends on the base learner.

The Capabilities

- **Class:** Data class, Numeric class, Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Date attributes, Missing values, Numeric attributes.

5.4.5. Lazy IBK

K-nearest neighbors classifier.

Can choose suitable value of K based on cross-validation. Moreover can do distance weighting [58].

The Capabilities

- **Class:** Data class, Numeric class, Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Date attributes, Missing values, Numeric attributes.

5.4.6. Lazy Kstar

K is an example of based classifier, which is the catagory of a best instance which is depend on the category of those training instances just like it, as figured out by some similarity function. It varies from other instance-based learners in this way that it uses an entropy-based distance function.

The Capabilities

- **Class:** Data class, Numeric class, Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Date attributes, Missing values, Numeric attributes[58].

5.4.7. Meta attribute selected classifier

Dimension of training and test data is decreased by attribute selection before being moved on to a classifier.

The Capabilities

- **Class:** Data class, Numeric class, Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Date attributes, Missing values, Numeric attributes.

5.4.8. Bayes net

Bayes Network learning makes use of numerous search algorithms and quality measures. It is base class for a Bayes Network classifier. It supplies data structure (network structure, conditional probability) distribution,….etc, and facilities common to Bayes Network Learning algorithms like K2 and B.

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Missing values, Numeric attributes[58].

5.4.9. Trees J48

Class for generating a pruned C4.5 decision tree:

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Date attributes, Missing values, Numeric attributes[58].

5.4.10. Naive bayes

Class for Naïve Bayes classifier makes use of estimator classes Numeric estimator precision values which are selected on the bases of analysis of the training data. For this very reason, the classifier is not an Updateable Classifier (which in typical usage are initialized with zero training instances), if one needs the Updateable Classifier functionality,one must use the NaiveBayesUpdateable classifier. The NaiveBayesUpdateable classifier uses a default precision of 0.1 for numeric attributes when building Classifier which is called zero training instances[58].

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Missing values, Numeric attributes.

5.4.11. Naive bayes Updateable

Class for Naïve Bayes classifier using estimator classes. The same classifier uses a default precision of 0.1 for numeric attributes when building a Classifier which is called zero training instances.

The Capabilities

- **Class:** Numeric class, Missing class values, Binary class.
- **Attributes:** Nominal attributes, Unary attributes, Binary attributes, empty nominal attributes, Missing values, Numeric attributes[58].

5.4.12. Naive bayes multinomial updateable

Catagory for constructing and utilizing a multinomial Naïve Bayes classifier, This can be explained as a class for constructing and utilizing a multinomial Naive Bayes classifier. The core equation for this classifier is:

$$P[C_i / D] = (P[D / C_i] * P[C_i]) / P[D] \quad \text{(Bayes rule)} \tag{5.3}$$

Where

C is class I, and D is a document, Incremental version of the algorithm.

The Capabilities

- **Class:** Nominal class, Missing class values, Binary class.
- **Attributes:** Nominal attributes[58].

CHAPTER 6
RESULTS

6.1. Experimental Results

The experiment here is done on the open source data mining tool of Weka. It was established at the University of Waikato in New Zealand, and the name presents for Waikato Environment for Knowledge Analysis. The program of system is written in C++. The algorithms can be implemented straight to a dataset. Weka has tools for data pre-processing, categorization, regression, clustering, association rules, and visualization. It is also well-applicable for further making new machine learning schemes.

6.2. Dataset

Our experiments were done on the datasets of Reuters-21578 Corpus. The collection is the most extensively used benchmark dataset for the text categorization research. We choice 10 categories of (acq, coffee, crude, earn, grain, interest, money-fx, money-supply, sugar, and trade). A (6078) as a total of training documents and (2677)test documents, which are selected for this study experiment and in our classification system.

6.3. Experiment Step

1) Document segmentation with computing technology, Reuters-21578 Corpus

2) Stop words, low-frequency words removed from the document, the rough dimension reduction.

3) Writing a program to represent the documents into document vector, feature weighting using the TF-IDF.

4) Document frequency evaluation function for feature selection.

5) The text features vector into the Arff format which Weka identifies and sparse data.

6) Arff file loaded into Weka, use Experimenter interface experiment and compares twelve text classification algorithms.

6.4. Evaluation Measure and Results

6.4.1. The Comparison between bayes and lazy classification algorithms

The following tables show the popular measures: Accuracy, Error Rate True Positive (TP), False Negative (FN), Recall, Precision and F-Measures. They are calculated for four learning machine algorithms Bayes Net, Naïve Bayes, Lazy.Kstar and Lazy.IBK.

Table 3 Experimental Results for Bayes Net and Naïve Bayes.

Algorithms		
Measurements	**BayesNet**	**NaiveBayes**
Accuracy	84.0705 %	75.67 %
Error Rate	15.93 %	24.33 %
TP Rate	84.1 %	75.7 %
FP Rate	3.2 %	1.9 %
Precision	83.8 %	81.6 %
Recall	84.1 %	75.7 %
F Measure	83.7 %	77.9 %

Table 4 Experimental Results for Lazy Kstar and Lazy IBK**.**

	Algorithms	
Measurements	**Lazy.Kstar**	**Lazy.IBK**
Accuracy	85.1199 %	84.8576 %
Error Rate	14.8801 %	15.1424 %
TP Rate	85.1 %	84.9 %
FP Rate	84.8 %	84.6 %
Precision	84.8 %	84.6 %
Recall	97.7 %	91.3 %
F Measure	79.85 %	79.53 %

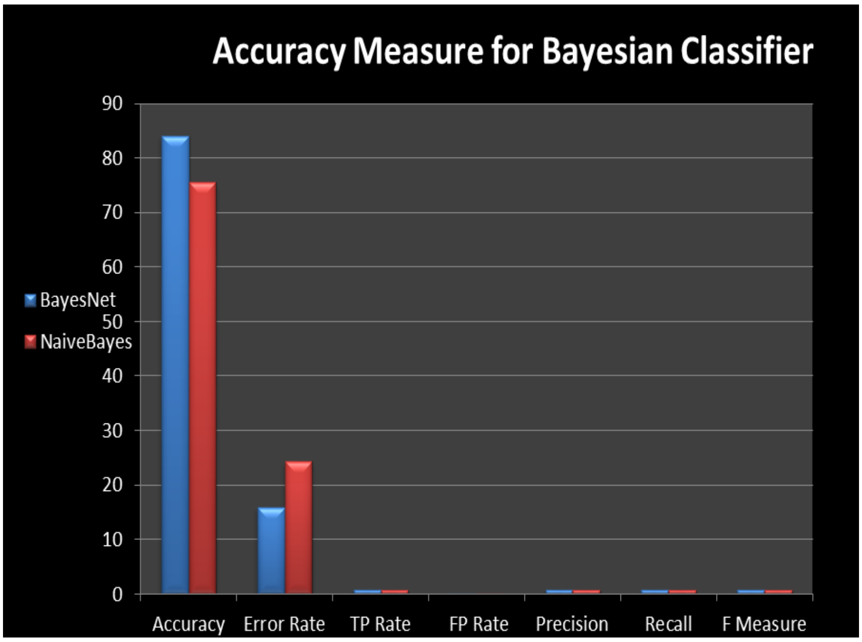

Figure 14 Accuracy Measures for Bayesian Classifier**.**

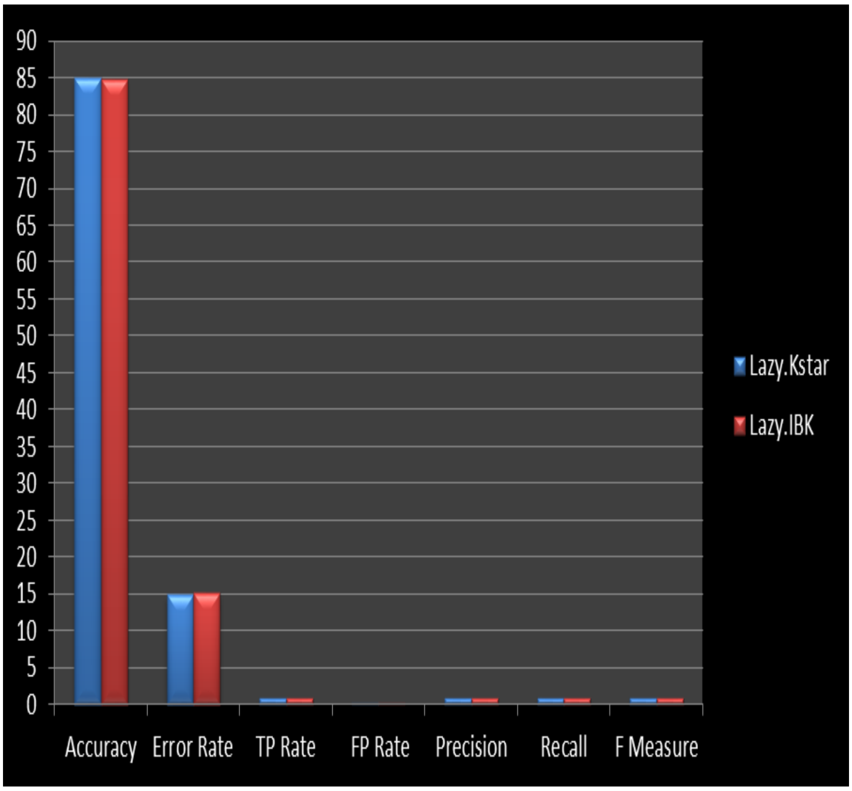

Figure 15 Accuracy Measures for Lazy Classifier.

From the analysis of Accuracy Measures of Bayesian Algorithm as shown in the table 3, Bayes Net performs well when compared to all accuracy measures namely, Error Rate, True Positive (TP), False Negative (FN), Recall, Precision and

F-Measures. As a result, Bayes Net outperforms well when compared to other Bayesian algorithm. From the Figure14, it is observed that Naïve Bayes attains highest error rate. Therefore, the Bayes Net classification algorithm performs well because it contains least of the error rate when compared to Naïve Bayes algorithm.

From the analysis of Accuracy Measures of Lazy Classifier as shown in the table 4, Kstar performs well when compared to all accuracy measures namely, Error Rate, True Positive (TP), False Negative (FN), Recall, Precision and F-Measures. As a result, Kstar outperforms well when compared to other Lazy algorithms. From the Figure 15, it is observed that IBK algorithms attains highest error rate. Therefore, the Kstar classification algorithm performs well because it contains least error rate when compared to IBK algorithms.

Table 5 Experimental Results for Bayes Net, Naïve Bayes, Lazy Kstar and Lazy IBK**.**

	Algorithms			
Measurements	**Lazy. Kstar**	**Lazy. IBK**	**Bayes Net**	**Naive Bayes**
Accuracy	85.12 %	84.86 %	84.07 %	75.67 %
Error Rate	14.88 %	15.14 %	15.93 %	24.33 %
TP Rate	85.1 %	84.9 %	84.1 %	75.7 %
FP Rate	3 %	3.3 %	3.2 %	1.9 %
Precision	84.8 %	84.6 %	83.8 %	81.6 %
Recall	85.1 %	84.9 %	84.1 %	75.7 %
F Measure	84.8 %	84.6 %	83.7 %	77.9 %

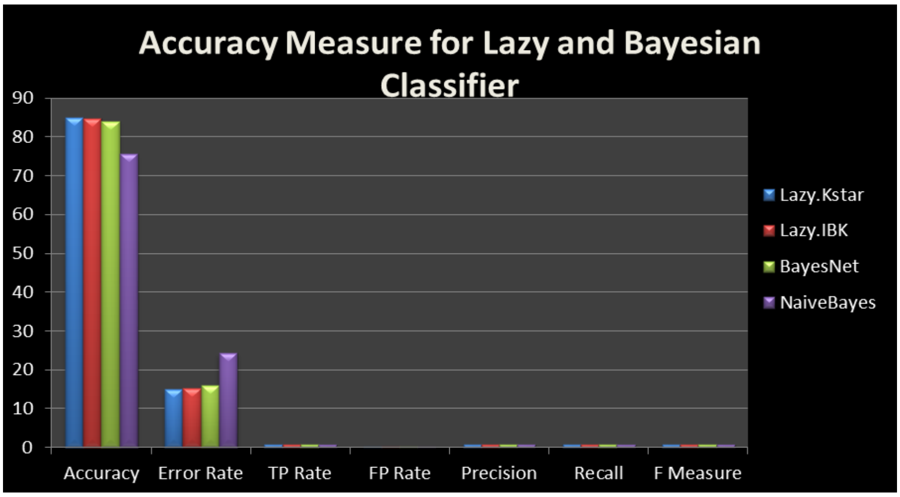

Figure 16 Accuracy Measure of Bayesian and Lazy Classifier.

From the Figure 16, it is observed that IBK and Kstar algorithms perform better than Bayesian algorithms. Therefore, the IBK and Kstar classification algorithms perform well because they contain highest accuracy when compared to Baye**s.** The Bayes Algorithm includes two techniques namely Bayes Net, Naïve Bayes and the Lazy algorithms includes IBK (K-Nearest Neighbor) and KStar techniques.

By analyzing the experimental results it is observed that the lazy classifier's KStar and IBK classification techniques have yields better result than other Bayes techniques.

Those results are coincided with results of [60], but the title of that article had a wrong, when it is making a comparative analysis of **(Bayesian)** and Lazy classification algorithms as in general, because it is proved in the next section that Naïve Bayes Multinomial (one type of a Bayes classification algorithms), it has yields better results than other techniques (Bayes Net, Naïve Bayes and the Lazy IBK and Lazy KStar) techniques.

6.4.2. Naïve bayes multinomial results

The table shows the popular measures: Accuracy, Error Rate, True Positive (TP), False Negative (FN), Recall, Precision and F-Measures. They are calculated for five learning machine algorithms Naïve Bayes Multinomial, Bayes Net, Naïve Bayes, Lazy.Kstar and Lazy.IBK.

The analysis of Accuracy Measures of Naïve Bayes Multinomial Classifier from the table 6, Naïve Bayes Multinomial performs well when compared to all accuracy measures namely: Accuracy, Error Rate, True Positive (TP), False Negative (FN), Recall, Precision and F-Measures. As a result, Naive Bayes Multinomial outperforms well when compared to other Lazy.Kstar, Lazy.IBK, Bayes Net, and Naive Bayes algorithms are respectively.

From figure 17, it is observed that Naive Bayes Multinomial algorithm attains highest Accuracy. Therefore, the Naive Bayes Multinomial classification of algorithm performs well because it contains least error rate when compared to Bayes Net, Naïve Bayes, and Lazy Kstar and Lazy.IBK algorithms.

Table 6. Experimental Results for Naive Bayes Multinomial, Bayes Net, Naïve Bayes, Lazy Kstar and Lazy IBK.

	Algorithms				
Measurements	**NaiveBayes Multinomial**	**Lazy. Kstar**	**Lazy. IBK**	**Bayes Net**	**Naive Bayes**
Accuracy	87.5562 %	85.12 %	84.86 %	84.07 %	75.67 %
Error Rate	12.44 %	14.88 %	15.14 %	15.93 %	24.33 %
TP Rate	87.6 %	85.1 %	84.9 %	84.1 %	75.7 %
FP Rate	2 %	3 %	3.3 %	3.2 %	1.9 %
Precision	87.7 %	84.8 %	84.6 %	83.8 %	81.6 %
Recall	87.6 %	85.1 %	84.9 %	84.1 %	75.7 %
F Measure	87.5 %	84.8 %	84.6 %	83.7 %	77.9 %

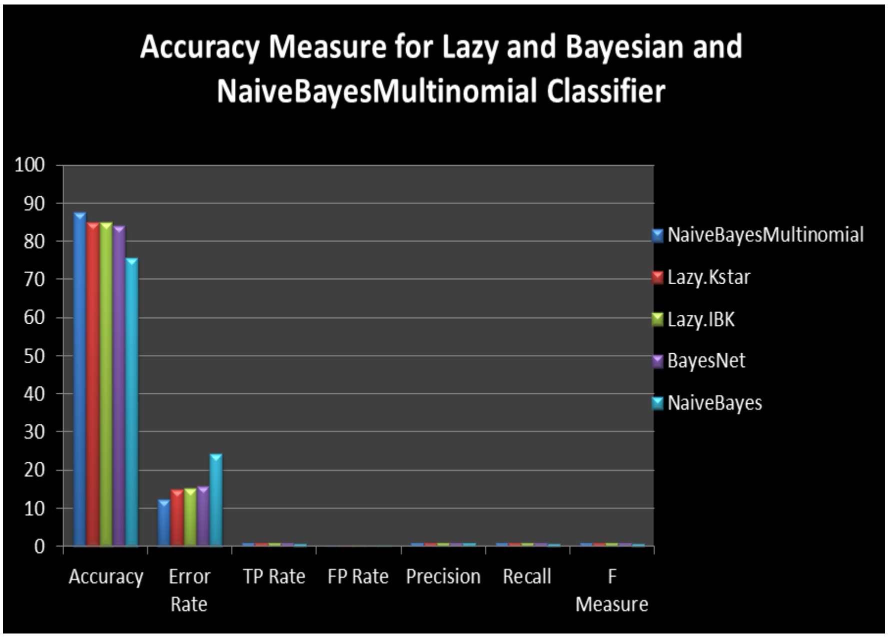

Figure 17 Accuracy Measures of Naive Bayes Multinomial, Bayesian and Lazy Classifiers.

6.4.3 Effectiveness threshold of document frequency on system performance

We determined document frequency's range to be from (100) to (1000). We, then, run a loop from (100) to (1000) and incremented the cutoff value by (100). Here, for example, cutoff value of 100 would mean that, a feature has to appear in at least (100) documents. In other words, we select only those features that are in (100) or more documents.

Below given are the sample results. Those results are represent the F-Measure and are obtained for range of Document Frequency (DF) from (100) to (1000), and for twelve learning machines algorithms (Naïve Bayes Multinomial,Sequential Minimal Optimization (SMO), TreesRandomForest, Meta.Bagging, Lazy.IBK, Lazy.Kstar, Meta.AttributeSelected Classifier, BayesNet, Trees.J48, NaiveBayes**,** NaiveBayesUpdateable, Naive Bayes Multinomial Updateable) respectively.

Table 7 shows the experimental results of F-M for Naïve Bayes Multinomial, Sequential Minimal Optimization (SMO), TreesRandomForest, and Meta.Bagging, and this table is followed by table 8 & table 9.

Table 7. Experimental Results of F-M for Naïve Bayes Multinomial, Sequential Minimal Optimization (SMO), Trees Random Forest, and Meta Bagging

Document Frequency (DF)	NaiveBayes Multinomal	Sequential Minimal Optimization (SMO)	Trees. Random Forest	Meta. Bagging
100	0.912	0.902	0.873	0.866
200	0.888	0.894	0.861	0.857
300	0.875	0.878	0.855	0.843
400	0.859	0.849	0.834	0.811
500	0.844	0.834	0.835	0.805
600	0.824	0.823	0.782	0.791
700	0.748	0.779	0.773	0.751
800	0.711	0.761	0.759	0.745
900	0.658	0.736	0.747	0.733
1000	0.624	0.72	0.741	0.702

Table 8. Experimental Results of F-M for Lazy IBK, Lazy Kstar, and Meta Attribute Selected Classifier and Bayes Net.

DF	Lazy. IBK	Lazy.Kstar	Meta.Attribute SelectedClassifier	BayesNet
100	0.858	0.846	0.846	0.843
200	0.861	0.856	0.824	0.842
300	0.846	0.848	0.811	0.837
400	0.833	0.839	0.789	0.834
500	0.827	0.826	0.77	0.822
600	0.813	0.816	0.742	0.82
700	0.785	0.784	0.706	0.787
800	0.773	0.771	0.712	0.777
900	0.745	0.765	0.696	0.756
1000	0.728	0.738	0.696	0.738

Table 9. Experimental Results of F-M for Trees J48, NaiveBayes**,** NaiveBayesUpdateable, and Naive Bayes Multinomial Updateable**.**

DF	Trees.J48	Naive Bayes	Naive Bayes Updateable	Naive Bayes Multinomial Updateable
100	0.835	0.797	0.797	0.771
200	0.826	0.788	0.788	0.688
300	0.805	0.779	0.779	0.633
400	0.788	0.774	0.774	0.575
500	0.785	0.766	0.766	0.529
600	0.766	0.767	0.767	0.472
700	0.735	0.742	0.742	0.438
800	0.729	0.728	0.728	0.321
900	0.71	0.721	0.721	0.259
1000	0.689	0.71	0.71	0.238

From the analysis of F-Measure of Naïve Bayes Multinomial Classifier from the tables (7, 8, and 9), shows Naïve Bayes Multinomial performs well when compared to F-Measures. The Naive Bayes Multinomial classification algorithm performs well because, it contains highest F-Measure when compared to Sequential Minimal Optimization (SMO), Trees Random Forest, Meta.Bagging, Lazy.IBK, Lazy.Kstar, Meta.AttributeSelected Classifier, BayesNet, Trees.J48, NaiveBayes**,** NaiveBayesUpdateable, Naive Bayes Multinomial Updateable) algorithms respectively.

It has got a good results related to the relation between document frequency and the F-Measure for twelve learning machines algorithms, From this Figure18, it is observed that Naive Bayes Multinomial algorithm attains highest F-M, and this factor decreases when increasing the number of document frequency, thus performance of the system deceases when document frequency increases. The following figures explain the performance of system (it is represented in F-M), and also for the twelve learning machines with respect to Document Frequency.

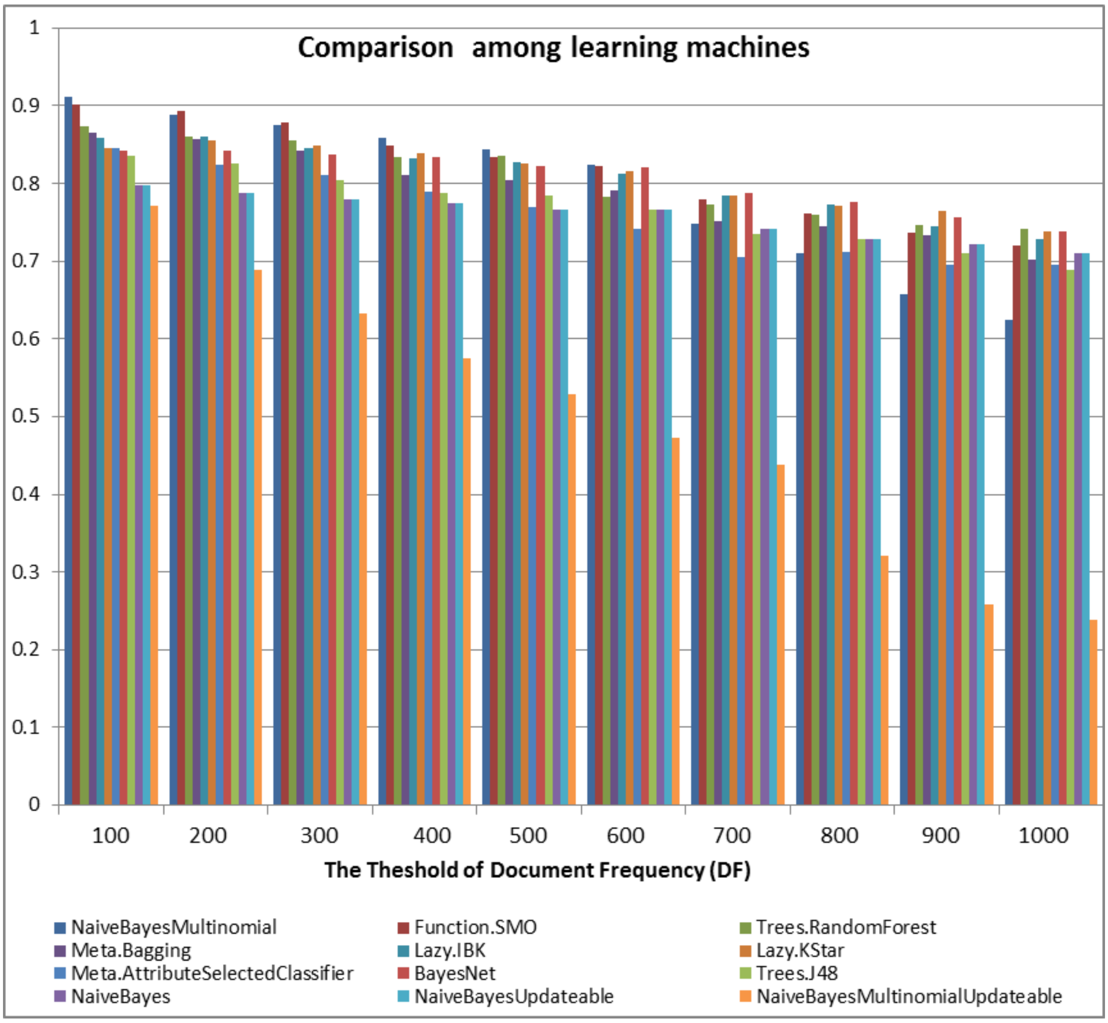

Figure 18 F-M measure of Naive Bayes Multinomial, Sequential Minimal Optimization (SMO), Trees Random Forest, Meta.Bagging, Lazy.IBK, lazy.Kstar, meta.attributeselected classifier, bayesnet, trees.J48, Naivebayes**,** NaiveBayesupdateable, Naive Bayes Multinomial Updateable.

From the Figure 19, it is observed that performance of system decreases when the document frequency increases, and it is observed that the Naive Bayes classifier's Multinomial classification technique yields better result than other techniques.

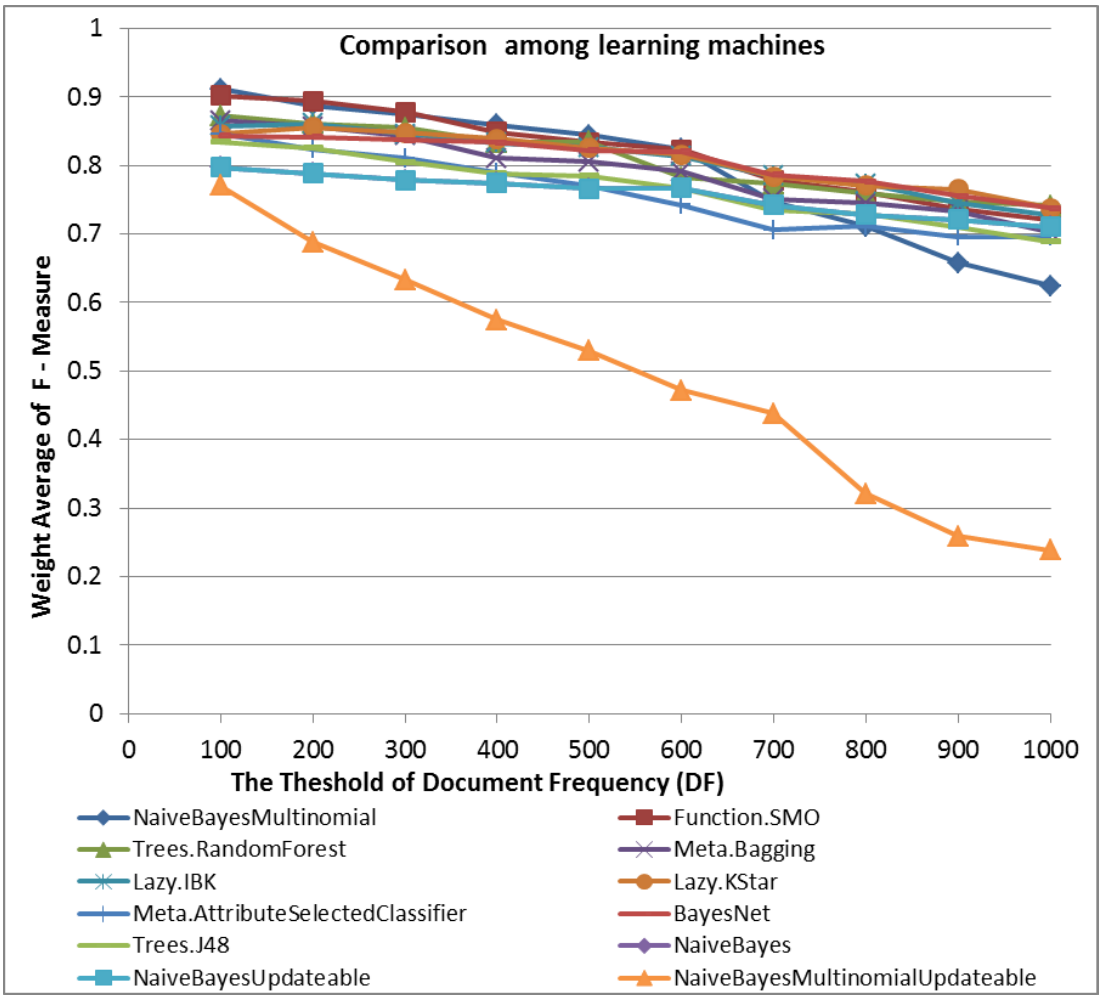

Figure 19 System Performances With Respect to Document Frequency.

6.4.4. Results for top 10 topics

We also calculated precision, recall, and F-M values for top 10 topics to determine how accurately documents were categorized when using different feature selection methods (tf-idf and DF).

F-M is calculated as follows:

$$F - M = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{6.1}$$

$$F - M = \frac{2PR}{P + R} \tag{6.2}$$

In this section, we preferred (tf-idf and DF) for top10 classes (earn, acq, coffee, crude, grain, interest, money-fx, money-supply, money-supply, sugar and trade). We fixed the size of features (19180) and document frequency equal to (100) and applied many learning machines algorithms separately. In the result, and from the tables (10, 11, and 12), we have found that F-Measure for (earn, acq,…. as example) class ,and the Naive Bayes Multinomial classification algorithm has higher when compared to Sequential Minimal Optimization (SMO), Trees Random Forest, Meta.Bagging, Lazy.IBK, Lazy.Kstar, Meta.AttributeSelected Classifier, BayesNet, Trees.J48, NaiveBayes**,** NaiveBayesUpdateable, Naive Bayes Multinomial Updateable) algorithms respectively for the same class. After that Lazy Trees Random Forest, Lazy.IBK, Lazy.Kstar Sequential Minimal Optimization (SMO)................etc. The following results were achieved, when we categorized (2677) test documents.

Table 10. Experimental Results of F-M for Top 10 Topics and for Four Agorithms below.

Topic	Naive Bayes Multinomial	Sequential Minimal Optimization (SMO)	Trees. Random Forest	Meta. Bagging
Earn	0.98	0.958	0.967	0.957
Acq	0.955	0.953	0.912	0.889
Coffee	0.863	0.783	0.638	0.873
Crude	0.873	0.892	0.811	0.884
Grain	0.888	0.897	0.832	0.899
İnterest	0.672	0.652	0.615	0.512
money-fx	0.748	0.751	0.672	0.623
money-supply	0.735	0.588	0.712	0.516
Sugar	0.732	0.759	0.56	0.724
Trade	0.748	0.769	0.703	0.706

Table 11. Experimental Results of F-M for top 10 topics and for Four Algorithms (Lazy.IBK, Lazy Kstar, Meta Attribute Selected Classifier and BayesNet.

Topic	Lazy. IBK	Lazy.KStar	Meta.Attribute SelectedClassifier	BayesNet
Earn	0.962	0.961	0.941	0.929
Acq	0.899	0.883	0.874	0.915
Coffee	0.553	0.419	0.918	0.651
Crude	0.735	0.669	0.851	0.72
Grain	0.797	0.793	0.865	0.825
İnterest	0.659	0.623	0.583	0.575
money-fx	0.627	0.686	0.525	0.614
money-supply	0.7	0.581	0.526	0.542
Sugar	0.603	0.552	0.783	0.708
Trade	0.684	0.67	0.637	0.645

Table 12. Experimental Results of F-M for Top 10 topics and for Four Algorithms (Naive Bayes Multinomial Updateable, Trees J48, NaiveBayes and NaiveBayesUpdateable).

Topic	NaiveBayes Multinomial Updateable	Trees. J48	Naive Bayes	NaiveBayes Updateable
Earn	0.937	0.938	0.944	0.944
Acq	0.846	0.85	0.871	0.871
Coffee	0.194	0.918	0.422	0.422
Crude	0.451	0.778	0.568	0.568
Grain	0.806	0.867	0.762	0.762
İnterest	0.142	0.502	0.409	0.409
money-fx	0.702	0.6	0.528	0.528
money-supply	0.211	0.513	0.319	0.319
Sugar	0.286	0.816	0.418	0.418
Trade	0.508	0.639	0.578	0.578

From this Figure 20, it is observed that Naive Bayes Multinomial algorithm attains the highest F-M for highest classes (earn, acq, coffee,……etc.), The following figures explain the behavior of the twelve learning machines with respect to top 10 topics of Reuters Corpus.

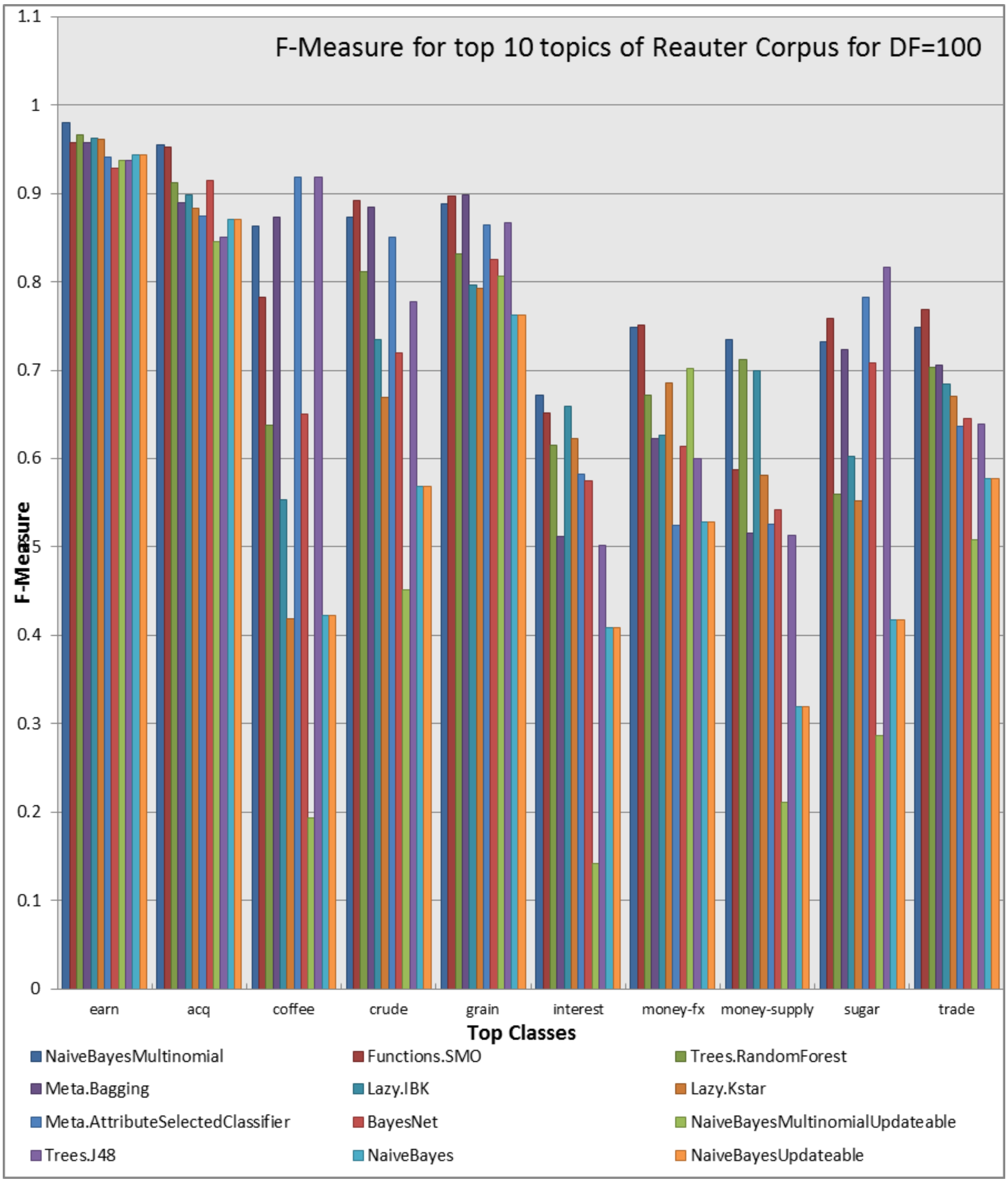

Figure 20 F-M Measure of Naive Bayes Multinomial, Sequential Minimal Optimization (SMO), Trees Random Forest, Meta Bagging, Lazy.IBK, Lazy.Kstar, Meta Attribute Selected Classifier, Bayesnet, Trees J48, NaiveBayes**,** NaiveBayes Updateable, Naive Bayes Multinomial Updateable with Respect to Top 10 Topics.

It explains the Micro-averaged F-measure, that provides equal weight to every document and hence it tends to be influenced by the classifier's performance on common classifications while reflecting the overall accuracy better. Precision and recall are obtained by summing over all individual decision:

$$\mathrm{Pr}\,ecision(P) = \frac{TP}{TP+FP} = \frac{a}{a+b} = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c} TP + FP_i} \tag{6.3}$$

$$\mathrm{Re}\,call(R) = \frac{TP}{TP+FN} = \frac{a}{a+c} = \frac{\sum_{i=1}^{c} TP_i}{\sum_{i=1}^{c} TP + FN_i} \tag{6.4}$$

Where C indicates the number of categories.

$$F - M = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{6.5}$$

$$Micro - averaged..F - M = \frac{2PR}{R+P} = \frac{2(TP)}{FP + FN + 2(TP)} \tag{6.6}$$

In the result, and from table.13, we have found that weight average for F-Measure for Naive Bayes Multinomial classification algorithm has the highest weight average when compared to Sequential Minimal Optimization (SMO), Trees Random Forest, Meta Bagging, Lazy IBK, Lazy Kstar, Meta AttributeSelected Classifier, Bayes Net, Trees J48, Naive Bayes**,** Naive Bayes Updateable, Naive Bayes Multinomial Updateable) algorithms, after that, SMO, Trees Random Forest, and so on.

Table 13. shows the results Weight Average for F-M with many Learning Machines.

Learning Machines Algoritms	Weight Average for F-Measure
Naive Bayes Multinomial	0.912
Sequential Minimal Optimization SMO	0.902
Trees Random Forest	0.873
Meta Bagging	0.866
Lazy IBK	0.858
Lazy Kstar	0.846
Meta Attribute Selected Classifier	0.846
Bayes Net	0.843
Trees J48	0.835
Naive Bayes	0.797
Naive Bayes Updateable	0.797
Naive Bayes Multinomial Updateable	0.771

From the figure 21, it is observed that Naive Bayes Multinomial algorithm attains highest weight average F-M,

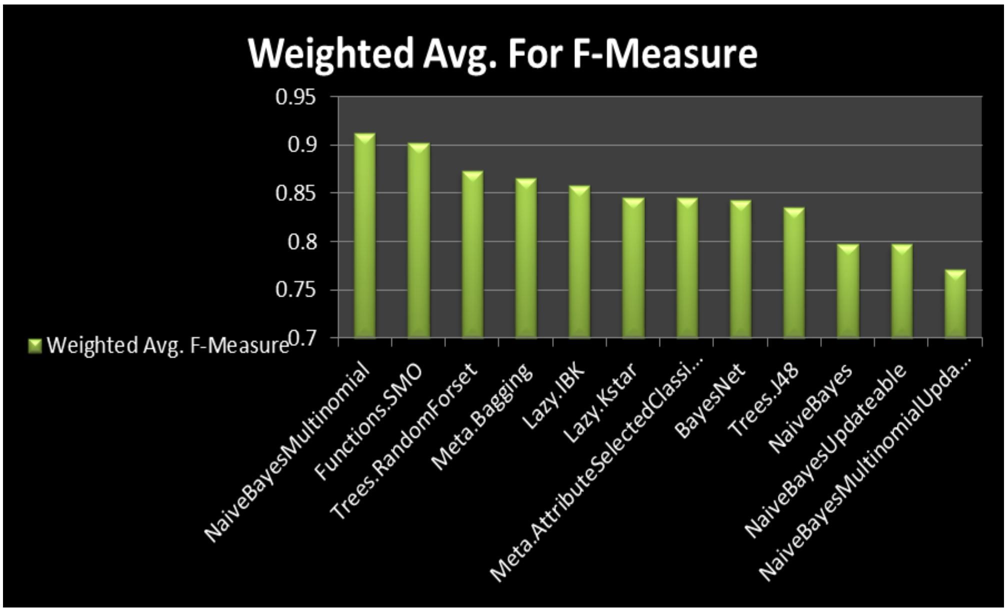

Figure 21 Weight average F-M Measure of Naive Bayes Multinomial, Sequential Minimal Optimization (SMO), Trees Random Forestt, Meta Bagging, Lazy IBK, Lazy Kstar, Meta Attribute Selected Classifier, Bayes Net, Trees J48, naive bayes**,** naive bayes updateable, naive bayes Multinomial Updateable for Top 10 Topics.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1. Conclusion

Text categorization is very important, but we believe, the problem of feature selection is equally as much, or more important than text-categorization. In this thesis, we discussed many important topics ranging from collecting data (training and test sets), to organizing data and ultimately using the organized data to efficiently conduct tests using the feature selection method, and we combined Term frequency-inverse document frequency *(tf-idf)* and Documents Frequency (DF) metrics to prepare the texts in a perfect way. After that, the same texts are used by classification process in Weka to get the best learning machines algorithms and the best performance of the system.

1- By analyzing the experimental results, to all accuracy measures namely: Error Rate, True Positive (TP), False Negative (FN), Recall, Precision and F-Measures it is observed that the Lazy Kstar classification algorithm performs well because it contains least of the error rate when compared to Lazy IBK algorithm, and Naïve Bayes attains highest error rate. Therefore the Bayes Net classification algorithm performs well because it contains least error rate when compared to Naïve Bayes algorithm. When we made a comparison between Bayes and lazy algorithms, it is observed that the lazy classifier's Kstar and IBK classification techniques have yielded better result than other techniques.

2- From the results, it is observed that Naive Bayes Multinomial algorithm attains highest Accuracy. Therefore, the Naive Bayes Multinomial classification algorithm performs well because it contains least of the error rate when compared to Bayes Net, Naïve Bayes, Lazy.Kstar and Lazy.IBK algorithms. Then, Naïve Bayes Multinomial (one type of a Bayes classification algorithms), it has yielded better results than other (Bayes Net, Naïve Bayes, Lazy IBK and Lazy KStar) techniques.

3- Results represent the F-Measure and are obtained for a range of Document Frequency (DF) from (100) to (1000), and for twelve learning machines

algorithms (Naïve Bayes Multinomial, Sequential Minimal Optimization (SMO), Trees Random Forest, Meta Bagging, Lazy.IBK, Lazy.Kstar, Meta. Attribute Selected Classifier, Bayes Net, Trees.J48, Naïve Bayes, Naïve Bayes Updateable, Naive Bayes Multinomial Updateable) respectively.
It is observed, that Naive Bayes Multinomial classification algorithm performs well because it contains the highest F-Measure when compared to Sequential Minimal Optimization (SMO), Trees Random Forest, Meta. Bagging, Lazy.IBK, Lazy.Kstar, Meta Attribute Selected Classifier, Bayes Net, Trees.J48, Naïve Bayes, Naïve Bayes Updateable, Naive Bayes Multinomial Updateable) algorithms respectively. It is observed that Naive Bayes Multinomial algorithm attains the highest F-M, and this factor decreases when increasing the number of document’s frequency, then performance of system deceases when document frequency increases.

4- From study results, it is observed that Naive Bayes Multinomial algorithm attains the highest F-M (precision and recall) for the highest classes (earn, acq, coffee,……etc.). It is explained the Micro-averaged F-measure, that gives equal weight to each document and therefore it tends to be dominated by the classifier's performance on common categories while reflecting the overall accuracy, we have found that weight average for F-Measure for Naive Bayes Multinomial classification algorithm has the highest weight average when compared to Sequential Minimal Optimization (SMO), Trees Random Forest, Meta. Bagging, Lazy.IBK, Lazy.Kstar, Meta Attribute Selected Classifier, Bayes Net, Trees.J48, Naïve Bayes, Naïve Bayes Updateable, and Naïve Bayes Multinomial Updateable) respectively.

5- Finally, the experimental results for (earn class as example),from those we concluded that Naive Bayes Multinomial classification algorithm has yielded better result than other techniques. It has F-measure equal to (98 %), and it is considered as the best algorithms for classifying Reuters-21578 data, after that the others algorithms are arranged as Sequential Minimal Optimization (SMO), and her (F-M=95.8%), Trees Random Forest, her (F-M=96.7%), Meta. Bagging, and her (F-M=95.7%), Lazy.IBK, and her (F-M=96.2%), Lazy.Kstar, and her (F-M= 96.1%), Meta. Attribute Selected Classifier, and her (F-M=94.1%), Bayes Net, and her (F-M=92.9%), Trees.J48, and her (F-M =93.7%), Naïve Bayes, 93.8%), Naïve Bayes Updateable, and her (F-M=94.4%), Naïve Bayes Multinomial Updateable, and her (F-M= 94.4%) respectively.

7.2. Future Work

1. In this thesis we focus on the combination between (Term Frequency-Inverse Document Frequency (tf-idf) and Documents Frequency (DF) metrics to prepare the texts. In future, we suggest combining the multiple feature selection metrics. Since, it is necessity to see the results of the combination of more than two methods in order to make a new conclusion.
2. Enhancement of the text classification methods such as (random forests support, vector machines (SVM), naïve Bayesian (NB), k-nearest neighbor (KNN), decision tree) Classifier for Text Categorization.
3. Planning is required to extend the studies with new feature selection metrics. Further future plan need to propose a model that will provide insight into which feature selection metrics can achieve better performance when combined.
4. In previous research, a text document is commonly represented by the term frequency and the inverted document frequency of each feature. Since there is a difference between important sentences and unimportant sentences in a document, the features from more important sentences should be considered more than other features which are not. We suggest measuring the importance of sentences. Then representing a document as a vector of features with different weights according to the importance of each sentence**.**
5. WordNet is a thesaurus for the English language based on psycholinguistics studies and developed at the University of Princeton . It was conceived as a data-processing resource which covers lexico-semantic categories called synonyms. It is possible to use WordNet for text categorization.

REFERENCES

1. **Pong J. Y., (2008),** "*A Comparative Study of Two Automatic Document Classification Methods in a Library Setting*", Journel Information Science, vol. 34, pp.213-230.

2. **Chidanand A.**, **Fred D., (1994),** "*Automated Learning of Decision Rules for Text Categorization*", ACM Transactions on Information Systems, vol. 12, pp.233-251.

3. **Fadi T., Mohammad A.**, **(2009),** "*Naïve Bayesian Based on Chi Square to Categorize Arabic Data*", Communications of the IBIMA, vol. 10, pp. 43-47.

4. **Youngjoong K. and Jungyun S.**, **(2000),** "*Automatic Text Categorization by Unsupervised Learning*", in Proceedings of the 18th Conference on Computational Linguistics, vol.1, pp.453-459.

5. **Philip J. and Steven P., (1990),** "*Construe-TIS: A System for Content-Based Indexing of a Database of News Stories*", in Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence, pp.49-64.

6. **Nobert F., Hartmann S. and Gerhard L., (1990),** *"AIR/X-a Rule Based Multistage Indexing System for Large Subject Fields",* in RIAO 91 Conference Proceeding: Intelligent Text and Image Handing, pp.6060-623.

7. **David L. and Marc R., (1994),** "*A Comparison of Two Learning Algorithms for Text*", Symposium on Document Analysis and IR, ISRI, Las Vegas, pp.45-49.

8. **Mohamed G., Mouloud K., and Oued S., (2013),** "*Arabic Text Categorization Using SVM Active Learning Technique*", Al Jouf University, Sakaka, Kingdom of Saudi Arabia, pp.50-54.

9. **Fabrizio S., (2001),** "*Machine Learning in Automated Text Categorization*", Consiglio Nazionale delle Ricerche, Italy, pp.52-54.

10. **George F., (2003),** "*An Extensive Empirical Study of Feature Selection Metrics for Text Classification*", Hewlett-Packard Labs, Palo Alto, CA., the Journal Machine Learning Research, vol. 3, pp. 1289-1305.

11. **Tin J. and Kwok Y., (1998),** "*Automated Text Categorization Using Support Vector Machine*", International Conference on Neural Information Processing (ICONIP).

12. **Salton G., Yang C. and Wong A., (1975),** "*A Vector-Space Model for Automatic Indexing* "*, Communications of the ACM*, vol. 18, pp. 613–620.

13. **Susan D., John P., Mehran S.,(1998), "** *Inductive Learning Algorithms and Representations for Text Categorization***",** Proceedings of the Seventh International Conference on Information and Knowledge Management, New York, NY, USA, pp.148-155.

14. **Yiming Y., Xin L., (1999), "***A Re-Examination of Text Categorization Methods* **",** Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.148-155.

15. **Joachims T., (1998).** *"Text Categorization with Support Vector Machines: Learning with Many Relevant Features"***,** Proceedings 10th European Conference on Machine Learning (ECML), Springer Verlag.

16. **Gungor T., (2006),** "*Classification of Skewed and Homogenous Document Corpora with Class-Based and Corpus-Based Keywords*", 29th German Conference on Artificial Intelligence (KI 2006), Bremen - LNAI (Lecture Notes in Artificial Intelligence), vol.4314, pp.91-101.

17. **Yang Y. and Pedersen O., (1997)** "*A Comparative Study on Feature Selection in Text Categorization*", Proceedings of the Fourteenth International Conference on Machine Learning, pp.412-420.

18. **Yiming Y. , (2000),** "*An Evaluation of Statistical Approaches to Text Categorization*", Kluwer Academic, Netherlands, pp.40-43 .

19. **Debole F. and Sebastiani F., (2003),** "*Supervised Term Weighting for Automated Text Categorization",* Proceedings of SAC-03-18th ACM Symposium on Applied Computing, ACM Press, pp.784–788**.**

20. **Özgür A. and Güngor T., (2005)** "*Text Categorization with Class-Based and Corpus-Based Keyword Selection*", Proceedings of the 20th International Symposium on Computer and Information Sciences, Lecture Notes in Computer Science, vol.3733, pp.607-616.

21. **Sherin S. and El-Sonbaty Y., (2007)** "*A Feature Selection Algorithm with Redundancy Reduction for Text Classification*", Proceedings of the 22nd Iinternational Symposium on Computer and Information Sciences, pp.312-316.

22. **Jan B. and Mohamed S., (2006)** "*Higher Order Feature Selection for Text Classification*", Springer-Verlag London Limited, Journal Knowledge and Information Systems, vol. 9, pp 468-491.

23. **Feng X., Tian J. and Liu Z., (2009)** "*A Text Categorization Method Based on Local Document Frequency*", IEEE, Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol.7, pp.468-471.

24. **Zhilong Z., Haijuan W. and Lixin H., (2011)** "*Categorical Document Frequency Based Feature Selection for Text Categorization",* IEEE, Proceedings of the International Conference of Information Technology, Computer Engineering and Management Sciences, vol. 2, pp 65 - 68.

25. **Hong Z., Yong R. and Xue Y., (2013)** "*Research on Text Feature Selection Algorithm Based on Information Gain and Feature Relation Tree",* IEEE, In 10th Web Information System and Application Conference, pp 446 - 449.

26. **Budiselic I., Delac G. and Vladimir K., (2014)** "*Developing a Text Classifier with Constrained Development and Execution Time",* International Conference of Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp 1170 - 1175.

27. **Nidhi G. and Vishal G., (2011)** "*Recent Trends in Text Classification Techniques",* International Journal of Computer Applications, vol.35, p.45-46.

28. **Fabrizio S., (2005)** "*Text Categorization",* Text Mining and its Applications to Intelligence, CRM and Knowledge Management.

29. **Mohamed H., (2007)** "*Automatic Documents Classification",* IEEE, International Conference of Computer Engineering & Systems (ICCES), pp.33-37.

30. **Jianlin C., (2011)** "*A Comparative Study on the Similarity Measurement of Text Categorization",* Master Thesis. Utah State University, Logan, UT.

31. **Miguel M., Alejandro B. and Thomas R. , (2013),** "*Document Difficulty Framework for Semi-Automatic Text Classification",* Proceedings of the 15th International Conference, (DaWaK), vol.8057, pp 110-121.

32. **Biebricher P., and Gerhard, (1988),** "*Automatic Indexing System AIR/PHYS Research to Application",* Research and Development in IR, Proceedings of the 11th Annual International ACM SIGIR Conference , pp 333-342.

33. **Jerry R. H., (1991),** "*SRI International: Description of the TACITUS system as used for MUC-3",* Association for Computational Linguistics, Proceedings of the 3rd Conference on Message understanding.

34. **Ralph G., John S. and Catherine M., (1991),** "*Description of the PROTEUS System as Used for MUC-3",* New York University , Association for Computational Linguistics, Proceedings of the 3rd Conference on Message Understanding, Morgan Kaufmann, pp 183-190.

35. **DeJong, G. F., (1982),** "*An Overview of the FRUMP System",* Lawrence Erlbaum Associates, Strategies for Natural Language, pp 149–176.

36. **Alessandro Z., (2005),** "*Text Mining and its Applications*", WIT Press, Southampton, UK, pp. 109-129.

37. **Lewis D., (1992),** "*Important Representation and Learning in Information Retrieval",*University of Massachusetts, USA, PH.D.

38. **Salton, G., Yang C., and Wong A., (1975),** "*A Vector-Space Model for Automatic Indexing",* Communications of the ACM, vol. 18, pp. 613–620.

39. http://ftp.cs.cornell.edu/pub/smart/, (Data Download Date: 12.05.2014).

40. **Porter, M. F., (1980),** "*An Algorithm for Suffix Stripping*", Program, vol. 14, pp. 130–137.

41. http://tartarus.org/~martin/PorterStemmer/ , (Data Download Date: 25.05.2014).

42. **Salton G. and Buckley C., (1988),** “*Term Weighting Approaches in Automatic Text Retrieval*”, Information Processing and Management vol.24, pp. 513–523.

43. **Zobel J. and Moffat A., (1998),** “*Exploring the Similarity Space*”, SIGIR Forum, vol. 32, pp. 18–34.

44. **Dasgupta A., Drineas P. and Mahoney M. W., (2007),** “*Feature selection methods for text classification*”, Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 07, vol. 21, pp. 230-239.

45. **Yanling Li, (2009),** "*Threshold Determining Method for Feature Selection*", IEEE Computer Society Washington, Proceedings of 2nd International Symposium on Electronic Commerce and Security, vol. 2, pp. 273-277.

46. **Fragoudis D., Meretakis D., and Likothanassis S., (2005),** "*Best Terms: An Efficient Feature-Selection Algorithm for Text Categorization,*" Jornal Know Information System, vol. 8, pp. 16-33.

47. **Kandarp D., (2011),** " *Study of Feature Selection Algorithms for Text Categorization*", University of Nevada, Las Vega (UNLV), Master Thesis, Number of Pages are 85
.

48. **Shang W., Shang H. and Wang Z**., **(2007),** *"A novel feature selection algorithm for text categorization,",* Journal Expert Systems with Applications, vol. 33, pp.1-5.

49. **Kolcz A. and Prabakarmurthi V., (2001),** *"Summarization as Feature Selection for Text Categorization,"* , Proceedings of the Tenth International Conference on Information and Knowledge Management, pp. 365-370.

50. **Rogati M. and Yang Y., (2002) ,**"*High-Performing Feature Selection for Text Classification*", Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp.659-661.

51. **Li S., Xia R., C. and Huang C., (2009),** *"A Framework of Feature Selection Methods for Text Categorization,* ", in ACL/AFNLP, vol.2, pp. 692-700.

52. **Rafael A. and Ceccatto H. A., (2000)**, "*Intelligent Document Classification*"**,** Journal Intelligent Data Analysis, vol.4, pp. 411 - 420

53. **Li, Y., Hsu D. F. and Chung S. M.., (2009),** "*Combining Multiple Feature Selection Methods for Text Categorization by Using Rank-Score Characteristics*", International Conference on Tools with Artificial Intelligence - ICTAI, pp. 508-517.

54. **Andrew M. and Kamal N., (1998)**, *"A Comparison of Event Models for Naive Bayes Text Classification*", Carnegie Mellon University.

55. **John C. P., (2007),** *"Fast Training of Support Vector Machines Using Sequential Minimal Optimization*", MIT Press, In Proceedings of the Advances in Kernel Methods - Support Vector Learning.

56. **Pall O., Jon A. and Johannes R.(2006)**, *"Random Forests for Land CoverC* ", J. Pattern Recognition Letters, vol. 27, pp. 294-300.

57. **Leo Breiman, (2001)**, *"Random Forests*", University of California, USA.

58. Waikato Environment for Knowledge Analysis(Weka), **(1999-2014)**, version 3.6.11, University of Waikato, Hamilton, New Zealand.

59. **Christopher D. Manning , Prabhakar R., and Hinrich S. (2008)**, *"Introduction to Information Retrieval* ", Cambridge University Press New York, NY, USA.

60. **Dr. S. Vijayarani and Ms. M. Muthulakshmi, (2013)**, *"Comparative Analysis of Bayes and Lazy Classification Algorithms* ", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol.2, pp.3118-3124.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: AL-GARTANEE, Asmaa
Date and Place of Birth: 13 March 1971, Iraq / Baghdad
Marital Status: Married
Phone: +90 5319138689
Email: asmaamuhamed1971@yahoo.com

EDUCATION

Degree	**Institution**	**Year of Graduation**
M.Sc.	Çankaya University, Mathematics and Computer Science / Information Technology Program, Ankara, Turkey.	2015
B.Sc.	University of Baghdad, Science College, Iraq, Baghdad	1994
Diploma in Computer	Technical Management Institute, Baghdad, Iraq	1991
High School	Al-Mustakbel Secondary School	1989

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2003-Present	Ministry of Scientific & Technology , Information Technology Company , Baghdad, Iraq**.**	Specialist
1998-2003	Al-Khuarzmi Company (Government Company), Baghdad, Iraq.	Programmer
1991-1994	Organization of Research and Development, Babel Company, Baghdad, Iraq.	Programmer

FOREIGN LANGUAGES
English, Advanced Arabic.

PROJECTS

- The Effectiveness of Feature Selection Metrics on the Text Categorization Performance, Cankaya University, Ankara, Turkey.

HOBBIES
Programming, Travelling, Reading Books.