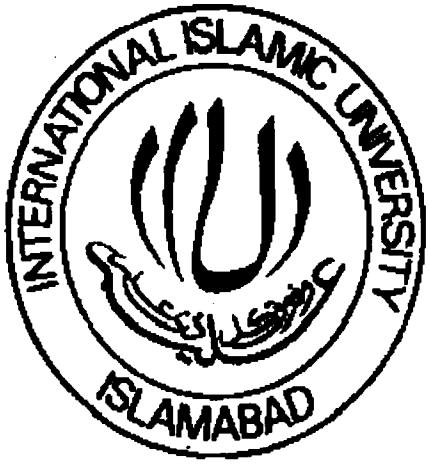
An Integrated Approach to Label Clusters

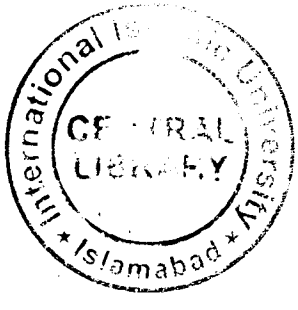


Developed By:
Asif Nawaz
(466-FBAS/MSCS/F08)

Supervisor
Dr. Rahat Hussain Bokhari
Director Computer Centre, QAU

Co-Supervisor
Syed Muhammad Saqlain
Assistant Professor, DCS, FBAS, IIUI

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad
2011



A Thesis Submitted to the

Department of Computer Science

International Islamic University Islamabad

as a partial fulfillment of requirements for the award of

the degree of

MS in Computer Science

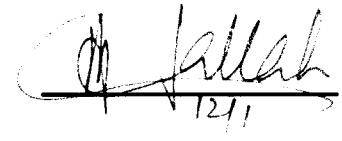
Dated: .....

Final Approval

It is certified that we have examined the thesis titled "An Integrated Approach to Label Clusters" submitted by Asif Nawaz, Registration No. 466-FBAS/MSCS/F08, and found as per standard. In our judgment, this research project is sufficient to warrant it as acceptance by the International Islamic University, Islamabad for the award of MS Degree in Computer Science.

Committee

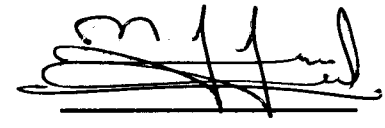
External Examiner



Dr. Nasro Min-Allah

HOD, Computer Science
Comsat Institute of Information Technology, Islamabad.

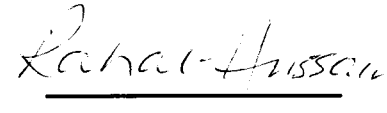
Internal Examiner



Imran Saeed

Assistant Professor
Department of Computer Science
International Islamic University, Islamabad

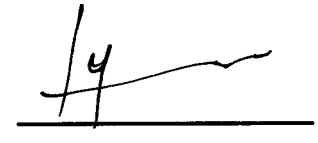
Supervisor



Dr. Rahat Hussain Bokhari

Director Computer Centre
Quaid-i-Azam University, Islamabad

Co- Supervisor



Syed Muhammad Saqlain

Assistant Professor
Department of Computer Science
International Islamic University, Islamabad

Dedicated To:

To my Kind Father

Who

Always Guide me to Learn and

Struggle for Learning More and More

Declaration

I hereby declare that this work, neither as a whole nor a part of it has been copied out from any source. It is further declared that I have developed the model, the software on the base of proposed model and the results with my personal efforts; and under the sincere guidance of Dr. Rahat Hussain Bokhari and Syed Muhammad Saqlain. If any part of this project is proved to be copied from any source or found to be reproduction of some other project, I shall stand by the consequences. No portion of the work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Asif Nawaz

(466-FBAS/MSCS/F08)

Acknowledgement

I simply bow my heads before Almighty Allah for giving me faith in my abilities and enabling me to accomplish this work and granting me with His Special Mercy, Blessings and Unlimited Help throughout the phases of the research work.

I conduct my profound regards to project supervisors Mr. Syed Muhammad Saqlain and Dr. Rahat Hussain Bokhari, who provided me their cooperation, guidance and valuable support during the all stages.

I would like to offer my special thanks to Dr. Sohail Asghar, Associate Professor MAJU, Dr. Daud Khattak, HOD Computer Science Department, AIOU and Dr. Ali Daud, Assistant Professor IIUI for their suggestion and comments.

At the last but not the least, I would like to adjudge acknowledgment to my family members especially to my uncle Mr. Imtiaz Ali Khan and Muhammad Johar. It surely would not have been conceivable without the orisons of my beloved mother and my friends specially Mr. Akhtar Aziz.

Asif Nawaz

(466-FBAS/MSCS/F08)

Abstract

Data Mining is the process of analyzing data from different prospective. Data Mining may also be defined as the process of extracting hidden patterns and valuable information from large text repositories. Various data mining techniques are used to perform different function on these data. Among all these techniques, one common technique is clustering. Clustering is the process of breaking of large record into smaller, similar and homogeneous groups. After the formation of groups called cluster these may be used by user for analysis and other purposes of interest. Such clustering chunks may be shaped in more useful way by assigning each of them and appropriate label/title to these documents and this process is called cluster labeling.

Cluster Labeling is the process of assigning appropriate and well descriptive title to a text documents. The most suitable label not only explains the central theme of a particular cluster but also provide means to differentiate it from other clusters in an efficient way. It assists the users to check whether a particular cluster contains the information relating to a particular needs/interest.

In this thesis we propose a novel technique for handling cluster labeling. The technique proposed will assigns a generic label that may/may not be a part of the document (text cluster) itself, it reflects the theme/idea induce debate in the document picked to be labeled. For assigning generic label to a document we adopted the use of external resources such as WordNet and thesaurus in generating alternative labels and then to choose most suitable one that may be capable enough to reflect the true essence of the text cluster to be labeled. Results are then compared with similar and existing techniques leading to the conclusion that the technique proposed in our research provides meaningful results.

Table of Contents

List of Figures & Tables

CHAPTER 1

INTRODUCTION

INTRODUCTION

1.1 Introduction

The procedure of searching and finding of valuable patterns in large databases is termed as data mining, knowledge extraction, information discovery, information harvesting, data archaeology, and data pattern processing [1]. The analysis of data from different prospective and the then to summarize it into valuable and useful information is called as Data Mining. Data mining process may also sometime call as data or knowledge discovery abbreviated as (KDD). By KDD we mean the discovery of useful knowledge or patterns from large databases, whereas data mining refers to a particular step in this process [2]. Through Data mining one can easily analyze, categorize, summarize and then identified valuable patterns inside a data. Data mining techniques are usually helpful in extracting hidden patterns from large block of data [2]. Technically, data mining process is used for identifying valuable patterns and correlation between different types of attributes inside large database repositories.

Data mining techniques/tools are helpful for both text mining and numerical data. Some of the common techniques used for text mining are pattern mining [3], subject-based mining [3], association rule learning [4], classification [5] and clustering [5]. Common techniques used for numerical data are regression [5], suffix tree method [6], co- relational coefficient [7] and Chi-square test [8]. Clustering is the process of breaking big records of data into smaller, similar, homogeneous groups [5]. Our research work is more concerned with text mining, specifically cluster labeling and its importance in the field of data mining.

1.2 Back ground

One common feature of text mining is the classification of data into similar and common groups. Clustering process partitions large text documents into smaller and similar groups. A well clustering technique performs grouping in such a way that it produces high inner group similarity and low outer group similarity [5].

Clustering is natural grouping or unsupervised classification of data into groups such that data objects similar to one another lies within the same cluster and dissimilar data objects in other clusters [7]. Clustering methods with better capabilities may produce high quality inside-cluster likeness and low outside-cluster likeness [7]. The excellencey of cluster depends on the measurement of similarity, used by the method and its implementation. The quality of a clustering method is also depends upon its ability to discover some or all of the hidden patterns [7].

After the formation of groups namely clusters, there exists a need to assign appropriate title to each cluster and the process is called cluster labeling. Moreover, labeling a cluster depends upon its specific characteristics for better understandability on the part of its use. It might help the users to understand the theme of the cluster formed without going through its detail.

1.2.1 Clustering Techniques

Different techniques have been developed and are in use to cluster similar data objects and label cluster formed so far. Teseng [7] devised hypernym search algorithm to produce generic cluster labels. In the first step contents indicative terms are extracted by using a chi-square test and correlation coefficient. Extracted terms are then mapped to their common hypernyms and then matched with database repository to get the general label. Popescul [8] presented two novel methods of labeling document clusters. The first method called as "frequent and predictive word methods" that select those words that occur both productively and frequently in that cluster and efficiently describe the given cluster, whereas in the second proposed method Chi-square tests of independence are performed to pick most frequent words. Carmel [9] improved a clustering labeling mechanism by using external resource Wikipedia. According to the proposed technique the given document is first indexed and then the documents are clustered by using various clustering techniques [5]. After the formation of clusters the proposed technique extracts important terms based on technique of Cutting and Karger [10] that describes the given cluster efficiently. Candidate labels for a given clusters are then generated by using important terms or Wikipedia. Then statistical co-occurrence and point wise mutual information [11] technique is applied to evaluate the labeling quality of each candidate term. The top candidate word selected by this

process is considered as label. Rizwan [12] proposed an algorithm called "EROCK" for clustering and topic generation. This algorithm consists of two steps: (i) Documents are converted into clusters by using cosine similarity measure, (ii) The most frequent word appearing in a given cluster is selected as label. Cluster labeling on the basis of frequent words selection may lend to poor performance in case the document contain wide range of knowledge [7].

Some limitations have been found in the past research regarding labeling text clusters and are mentioned as under.

- There exist many manual labeling techniques. Although they are justifiable but they are costly and time consuming.

- Various techniques in practice for text labeling select the label which is not almost generic. The labels selected are often part of the document itself which may not reflect the theme strongly especially when the document encompasses wide range of related/relevant aspects.

- The existing techniques hardly provide efficient and meaningful labels that may help the users to quickly identify clusters of interest without examining particular documents in detail.

- Particularly for text clustering and labeling the statistical equations used for labeling a cluster, provide results (i.e. label of cluster) that may not be considered as a good cluster descriptor.

Such limitations need to be addressed leading to research question mentioned in the next section.

1.3 Research Questions

1. What type of label may be considered as a good label for a given cluster and why?
2. Can "frequent item set selection" technique be improved?

3. Can better text label be selected without using complex statistical equations?

1.4 Objective

The main objective of this research is to extend/improve text labeling technique to provide a generic label that can efficiently describe the given cluster without discussing the particular cluster in detail. The technique proposed may provide an efficient and meaningful label by using an external resource like WordNet. Our study aims to enhance the existing body of knowledge about the existing labeling techniques for developing more refined/suitable labels. Objectives of the proposed research are as under:

- To enhance the existing text labeling techniques to be more efficient.

- To generate a generic label capable enough to describe the whole cluster rather than its part.

- To develop alternative text labeling techniques that may not include complex statistical equations to generate a label.

1.5 Scope of the Research

The proposed model is extremely successful in the field of text clustering and labeling. Following are the some bullets that cover the scope of the proposed research work:

- It may facilitate the data mining researchers and analysts to search their topic/ clusters quickly and efficiently.

- It may provide a simple method of allocating a good label instead of using complex equations and methodologies.

- It may develop a generic method of allocating label that described the given cluster even if it contains a wide range to text information.

- It may overcome shortcomings of up to 65% of the existing technique for text labeling.

1.6 Research Framework

An incorporated technique based on outer sources like WordNet is presented to resolve the difficulty of the weak readability of cluster description. The proposed technique creates general labels for efficient cluster interpretation. It label cluster automatically rather than using manually labeling technique [13].

It may provide generic titles for clusters. The task will be accomplished in four phases described as follow and shown in figure 1.1.

- Clustering: That makes clusters of given data set.

- Stemming: In this phase common words like full stop, commas, articles, pronouns etc are removed.

- Term Extraction: Frequency is calculated for each of the remaining word in a cluster, term with the highest frequency is selected for next phase.

- Extraction Refinement: Frequency of the candidate word is further refined using treasures of that word in that cluster, and consequently updating the frequency of word already calculated. The word with highest frequency is again selected as a candidate label.

- Labeling: Compare all candidate label with WordNet to calculate generic label leading to the final label of the corresponding cluster

1.7 Chapters Breakdown

Let us talk about how the remaining chapters of the thesis have been structured. Chapter number 2, titled literature Review, describes the past work/research related to the proposed work. In Chapter 3 i.e. Cluster Labeling review the definition and concept of cluster labeling and also focus on basic need and importance of cluster labeling in field of data mining. Chapter 4 discusses the proposed conceptual model design. Chapter 5 shows the experimental results observed by the software designed on the base of our proposed model. Chapter 6 describes the conclusion and future work regarding to the field of study. Second last section describes the research paper, and last section appendices are given. The detail of chapter's breakdown is shown in Figure 1.2.

CHAPTER 2

Literature Review

LITRATURE REVIEW

An amazing progress in database technology provided a great boost for fast growing data to be collected and stored in large database repositories. There exists huge volume of scientific, marketing, medical, financial data that has for executed the human capability to analyze it without data analysis tool. Moreover, g overnment organizations, engineering and scientific institutions, and businesses firms spending enormous resources to gather and store data and further analyze it for their strategic decision making [1]. However, in reality, only a specific amount of such data will ever be used because the decision makers do not have tools available to them for extraction of knowledge embedded in such large and numerous data repositories [1]. In various situations the amount of data is so large to be managed as the data structures itself are complex. To analyze such huge volume of data in an efficient way is a challenging task. The main cause may be that much focus was given to issue of storage efficiency during the creation of data rather than planning of how data at the end to be analyzed [14].

The need to understand huge, complex, informative data sets is apparent to compete in such a competitive environment. For example in finance and business, corporate and customer data are considered as a strategic property. To pull out valuable information hidden in data and, and its further use is almost essential in today's challenging world. The overall process of introducing computer based methods along with new technique for discovering such hidden information and knowledge is known as data mining [14]. The efficient and effective analysis of data available in different forms always needs effective techniques for analysis and summarization of data.

2.1 Data Mining

Data mining is defined as the Discovery of knowledge either by automatic method or by manual method [15]. Data mining is most helpful in an exploratory study situation

in which someone have no predefined ideas about what will be the outcome [15]. The process of searching new, valuable, and nontrivial information in huge and complex repository of data may be termed as Data Mining [14].

Through data mining techniques we can extract convincing, previously unknown, comprehensible, and actionable information from large text repositories of databases and using it for making strategic management and business policies. Data mining helps to extract hidden patterns from large block of data [3] and analyze it from different perspectives and summarize it into useful information [3]. The procedure of searching and finding of valuable patterns in large databases is termed as data mining, knowledge extraction, information discovery, information harvesting, data archaeology, and data pattern processing [1]. Data mining may sometimes also be called as data or knowledge discovery abbreviated as KDD but By KDD we mean the discovery of useful knowledge or patterns from large databases, whereas data mining refers to a particular step in this process [2] as shown in Figure 1 [3]. Through Data mining one can easily analyze, categorize, summarize and then identified valuable patterns inside a data. Data mining techniques are usually helpful in extracting hidden patterns from large block of data [2], data mining techniques are helpful in identifying association or patterns existence between various attributes of entities involved in large relational database repositories [3].

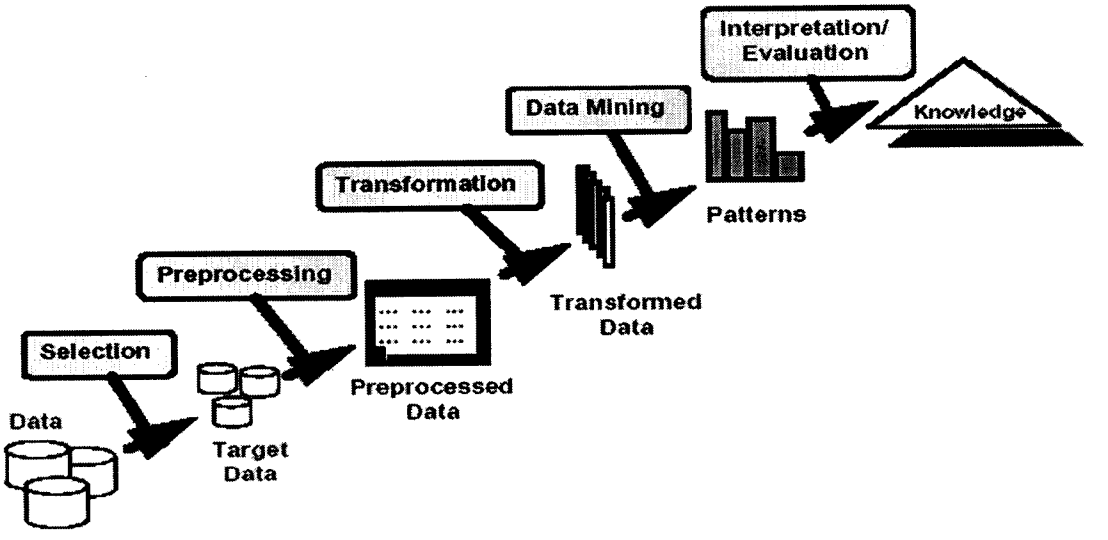


Figure 2.1 An Overview of the Steps of the KDD Process [3].

Two major tasks of data mining are considered as "prediction "and "description "[15]. Predictive data mining identifies future values using some variables and the model of the system described by the given data set [15], while on the other hand descriptive data mining is concerned with pattern finding that may be helpful for users to perform operations on data [15]. Data mining technique are used to achieve the goal of both description and prediction.

Text Mining (TM) is the procedure used to pull out new, unobserved and unstructured information secreted in large repositories of unstructured text databases, using advanced technology [16]. Through text mining one can uncover correlation in a text files and text data and to search them in order to get new knowledge/patterns. Text Mining is getting more importance day today because of the huge collection of data and knowledge that resides in text databanks available online, within the enterprise, elsewhere, or any combination of these sources [16].

2.2 Data Mining Techniques

There exist advanced and well developed tool and techniques related to data mining in order to identify hidden knowledge, unknown patterns and relationships in transactional data and databases based on user demand [14]. Some of the most common data mining techniques are briefly described in the next section.

2.2.1 Association Rule as an unsupervised technique of data mining is used to identify relation between records in data [4]. It is also referred as market base analysis [5]. Association rule mining is a technique for identifying unsuspected data dependencies. Through association rule one can identify all the possible existing rules which satisfy some condition of user interest.

2.2.2 Pattern mining is another well known data mining technique that is used to search existing patterns [5]. The idea of pattern mining emerged from association rule in which original impulse for searching patterns came from the need to analyze market based transactional data, that is, to observe the customer behavior in terms of buying a product from the market. For example, associations rule "bred — butter (80%)" states that four out of five customers that bought bred also bought butter.

2.2.3 Regression is a data mining technique which is based on prediction [5]. Regression is used to predict any number, loss or profit, sales or purchase, house values, distance, temperature, or distance etc. For example, a regression model could be used to guess the worth of a house on the bases of its location, number of rooms and other factors etc.

A regression task starts with an existing data set in which the required values are known. For example, if someone wishes to predict his property value then the regression model predicts the property value by observing data form previous documentation/records of any property over a period of time.

2.2.4 Classification is a data mining (supervised learning) technique in which one develops a descriptive model for any known data [17]. Classification is helpful to calculate group membership for data instances [17]. Classification technique is most commonly used in commercial data mining. Using classification, different organization may discover patterns to solve complex business problems [5]. There exist many classification techniques/methods such as decision trees that split a data set and build a model which is used to classify each record in term of a target field [17]. Neural Networks on the other hand are also considered as popular method with the capability of finding patterns, and now successfully applied in classification [18]. Genetic Algorithm is another optimization technique that solves problems by copy the same processes of descriptive model [19].

2.2.5 Clustering is the process of breaking large record of databases into smaller, similar and identical groups [5]. Clustering is also a natural grouping or unsupervised method that classify the data in such away that group items of one group didn't match with the group items of other groups. The data objects having similar properties lies within one cluster [8]. A good clustering technique possibly produces clusters with high inside-cluster likeness and low outside-class similarity [14]. The excellency of clusters depends on the measurement of similarity, used by the method and its implementation. The quality of a clustering method is also depends upon its ability to discover some or all of the hidden patterns [7].

2.2.6 Cluster Labeling: There exists a need to label the cluster depending upon its specific characteristics for the better understandability on the part of its use. Labeled clusters improve the readability and help users to find out whether one of the clusters is relevant to user's requirement or not. Therefore, cluster labeling is getting more and more attention in the field of data mining. Regarding to the importance of cluster labeling different tools and techniques are developed. In this sections we overview of the well known existing techniques:

(i) Carmel [9] improved a clustering labeling mechanism by using external resource Wikipedia. According to the proposed technique, the given documents are first indexed and then clustered by using various clustering techniques [5], [10]. After the formation of clusters, the important terms are extracted [9] that describe the given cluster. Afterwards related Wikipedia information are extort from Wikipedia site. The final labels are then selected by using statistical co-occurrence point wise mutual information [11] and. A general framework of the described technique is given as shown in figure 2.2.

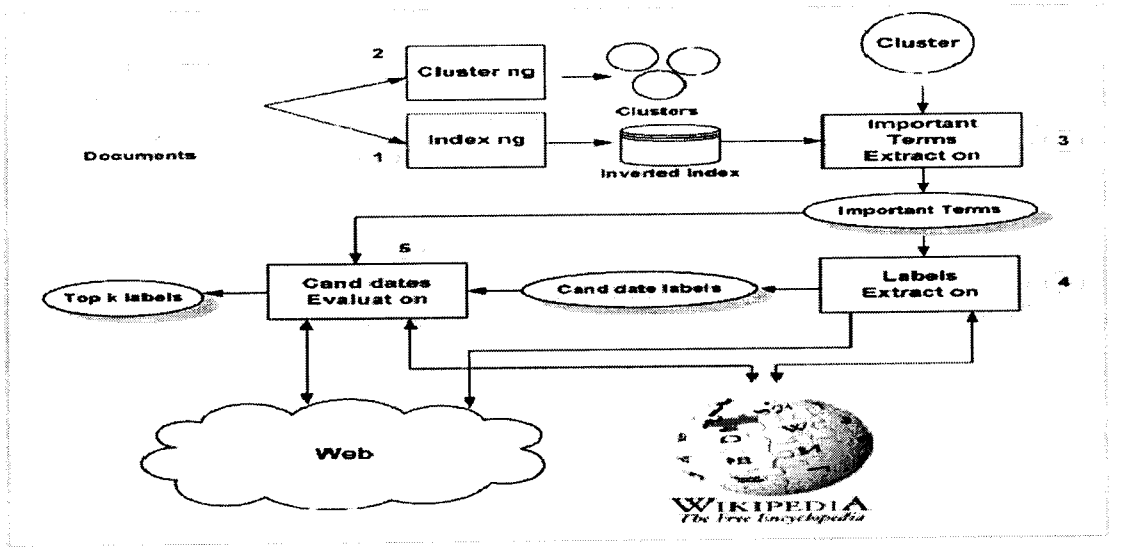


Figure 2.2d Technique of Labeling using Wikipedia adopted from [9]

It's a successful way to label those document whose information are available in Wikipedia. For the topics whose related information are rarely available in Wikipedia,

may effect the performance of the discussed technique. In order to make efficient use of Wikipedia, there is a need to take an intelligent decision to focus on inner term labeling.

(ii) Treeratpitul [20] proposed a labeling algorithm that label hierarchical clusters. The main idea of the proposed technique was taken from the concept of term frequency [28]. According to this technique If **S** is a cluster and **P** is their parent clusters which consists of the entire document in **S** and sibling clusters of **S,** the technique select labels for each cluster **S** in the following way.

(a) Collect Phrase Statistics: initially for each phrase that appears in the clusters whether it is unigram phrase, bi-gram phrase or tri-gram phrase, Document frequency (DF) which is the number of document in the cluster that contains that phrases and then Term Frequency (TF) which is the total number of occurrence of Phrases in the cluster is calculated.

(b) Select Label Candidates: The document frequency calculated in the above step is also used in the selection of labels. The threshold for a unigram phrase that occur is 20% of documents in the cluster, and bigram or trigram phrase if appear 5% of the document in a cluster will be consider as good descriptor.

(c) Descriptive Score (DScore): The aim of this step is to calculate the descriptiveness of candidate word. Label are arrange according to the descriptive score value. The DScore for any phrase is based on different features which are describe as:

- Normalized Document Frequency which can be calculated as:

$$\text{Normalized DF} = DF/|C|.$$

A good label will be consider that one whose term frequency is high in self cluster as compare to their parent cluster or other clusters.

- Term Frequency and Inverse Document Frequency which can be calculated as:

$$TFIDF = TF * \log(|C|/DF).$$

A phrase whose TFIDF value is high is considered as a good descriptor. With the help of this technique we can easily calculate TFIDF for parent and self cluster.

- On the bases of the above feature for every candidate label four ranks are calculated. And the candidate labels are sort according to its DS score. The label whose value is highest is assigned rank 1, represented as $(DFs/|S|) = 1$ so on the

procedure continue. The title will be good if it has high rank in parent and very high rank in self.

(d) Cutoff Point: The final step that point out that show the number of label candidates is to be displayed on the bases of DScore calculations.

The main strength of this technique is that it not only focuses on single word label but also calculate bigram label when required. However, this is a statistical approach and fails to generate label for large textual data.

(iii) Popescul [8] proposed two separate techniques namely "frequent and predictive" words method for labeling hierarchical clusters. Both were based on word selection that is used for labeling clusters.

(a) In first technique Chi-square test is performed on dataset. According to this technique for all nodes containing bags of words inside a cluster tree, Chi-Square test is performed. It calculates dependencies between words. The term dependencies mean that whether the word is occurring equally like in all of the children or not. If a test concludes that the word has different chances of appearing in its children nodes then it refers that the word is specific. If the test concludes that the word is equally likely occurring in all of the children then retain the word in present node and remove from all other nodes. The Chi-Square test is performed upto the leaves inside a cluster tree and its output will be a unique words that will be specific for a particular node and will not exists in the sub node of that present node. A label will be the most frequent word at the node corresponding to a cluster of document we want to label.

(b) In the second technique, the label is selected on the basis of the product of local frequency which means the existence of a particular word in a cluster, and productiveness which is a mutual information estimator explained in previous papers [21]. Although they are valid and descriptive, however none of this method gave satisfactory results in the internal node of the hierarchy.

(iv) Maqbool [22] describes the clustering process and clustering labeling in term of software context and software recovery. Two algorithms i.e. complete algorithm and weight combined algorithm are used while cluster labeling can be used to easily interpret the sub-systems of any software. The main method which is used for labeling is based on function identifiers of software. These function identifiers are used as a

representative key words for any entity and then Inverse Document Frequency (IDF) scheme is used to select an efficient keyword that lead to the label of that cluster (sub-system of software). The labeling process is the part of clustering process because labels are assigned to each cluster during clustering process. However, the method explained can be broken down into clustering process and labeling process.

- Clustering process: It first selects the entity and its features on the basis of global variables, local variables and user defined types used by an entity. Secondly, selection of similarity measures, then selecting clustering algorithm, and finally selection of evaluation method.
- Labeling Process: The procedure used for labeling is:

1) Label Selection: Since this method is entity based so the keywords of an entity will be selected as label.

2) Label ranking: After selection of multiple keywords the weight is calculated for each keyword using either of frequency or inverse document frequency. The keyword with highest weight is selected as label for that sub system.

(v) Tseng [7] proposed an algorithm named as hypernym search algorithm for cluster labeling. The proposed algorithm assign generic label to a particular cluster by using WordNet. Initially by performing chi-square tests and correlation coefficient high frequency and well descriptive words are extracted. Then hypernym search algorithm is applied to matched these words with WordNet to get final labels. Experiments are performed on text collection of Reuters-21578. Results so for obtained showed that the performance of this technique is outstanding. The proposed technique label clusters efficiently but it may have two shortcomings, (a) it contains a complex statistical equations and secondly, (b) it is unable to label those topics that are not addressed by the WordNet. This method can be easily extended to use other hierarchical resources such as database repositories, Babylon dictionaries for adaptable label generation.

(vi) Zhang [31] proposed an integrated approach based on Description Comes First (DCF) and Description Comes Last (DCL). Both were combined to generate cluster descriptor words. The proposed technique initially extracts high frequency keywords for each document by using description come first method [31]. The important terms

are also extracted on the bases of description come last method. Results are merged using the similarity measure technique. The word with highest value may also be called as highest score word and may be selected as label of cluster.

(vii) Zhang [13] described a cluster label selection technique using statistical machine learning. In this technique descriptive words are first ranked and then separated as descriptive and non descriptive words. For label generation three different type of machine learning method i.e. support vector machine model (SVM), Multiple Linear Regression (MLR) and Logistical regression model (Logit) are used [13]. Labels are selected by calculating various features of a particular word like global feature [13], local feature [13]. In the concept vector, the words with top five weights will be selected as a cluster descriptor while in heuristic method the importance (weight) of candidate labels are measured. The described technique may not obtain pleasing result there is a need to search more efficient clustering labeling features to improve the worth of clustering description [13].

(viii) Richard Fulton [33] proposed a paper that basically performs comparison between several clustering algorithms on the bases of semantic role labeling. For this purpose a baseline system based on logistic regression classifiers, and distributional clustering algorithm is used. It uses a Latent Semantic Analysis system to evaluate two formerly implemented clustering algorithms, k-means and a more comprehensive discriminative clustering algorithm. This research mainly focuses on features extraction through specialized corpuses such as WordNet. Overall, it describe that the performance of semantically text clustering leads to significant improvements on semantic role labeling tasks.

(viii) Rizwan [12] present an algorithm named as EROCK to make a cluster and then assign a labels to the clusters. According to this approach initially text files (data set) is arrange into text documents. Then cosine similarity [24] approach is applied to calculate association between all documents. After computing links between each document, each document is considered as a cluster. Clusters strength is reduced by merging two or more clusters to single cluster by using goodness measure [12]. After the formation of cluster the next phase is labeling the clusters. In this technique

clusters are labeled on the basis of term frequency method [7] and the highest frequency is selected as a label for that cluster.

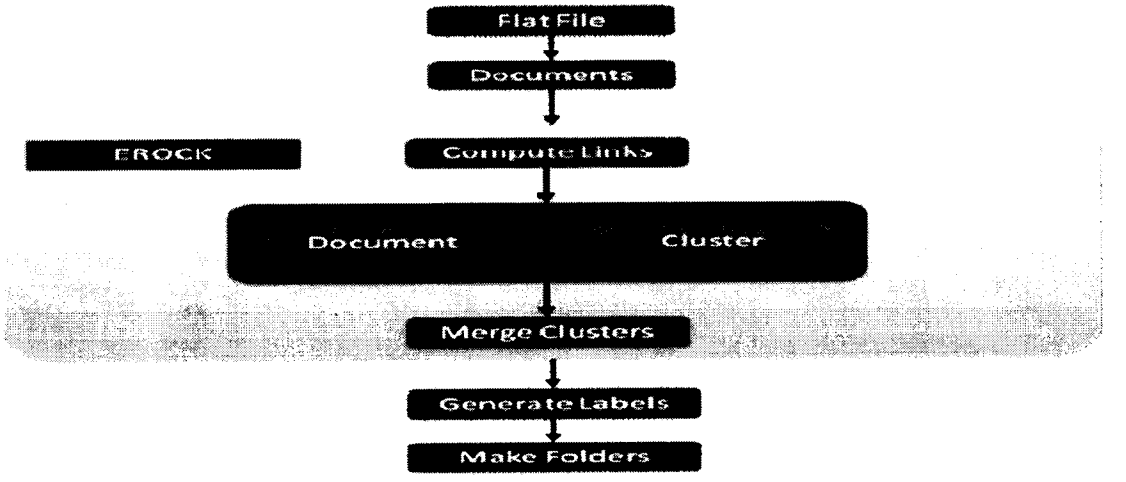


Figure 2.3Flow diagram of document topic generation with EROCK [12]

The major shortcoming of this approach is that topic title selected on the basis of repetitive word selection approach may lead to poor performance when the document contains huge text information [7]. Secondly, the label selected on the basis of term may not properly describe the given cluster [7].

(ix) Zhai [23] described a probabilistic method of automatic labeling of Multinomial topic models. Extract labels which are understandable, semantically related and efficiently describe the cluster, initially candidate labels are generated. In this method phrases are consider as a candidate label instead of a single word. For phrase generation two approaches are used.

(a) First approach called as chunking or shallow parsing, it selects that phrase or chunk as a candidate label which is most frequent in a particular cluster.

(b) The second approach called as N gram testing, it selects the candidate label by using statistical equations. In this method statistical equations are used to identify co-occurrence between words [5]. Then, different methods like mutual information [21], chi-square tests [7] is applied to check whether N gram is meaningful or not. After the candidate label extraction, two methods are used to rank each candidate label. The first method called as zero-order-relevance which consider that label as a good label that contain more important words and have some sense [23]. The second method called first-order-relevance which rank candidate label on the basis of semantic

relationship of candidate label with that cluster [23]. After ranking the candidate label the candidate label with highest label is considered as label for that cluster.

The proposed method is based on pure statistical equations and there exist a room to improve the quality of label by introducing better technique to select candidate labels and to rank candidate labels [23]. Secondly, this method labels simple topic models and there is a need to improve the technique to generate labels for hierarchical topics [23].

After studying the previous work we came to the conclusion that there are limitations in the existing research addressing cluster labeling. Most of the existing research regarding labeling a cluster is not generic. The label is picked from the document themselves which may not be enough particularly when the document (cluster) occupy huge text. Secondly, the existing techniques have limitation to generate efficient and meaningful labels that may help users to quickly recognize clusters of user's interest without investigating particular documents in detail. In most of the cases, pure statistical equations are used for labeling a cluster having difficulties in identifying an efficient label for a cluster. There also exist manual labeling techniques. Although they are good but they are expensive and time consuming [13]. It usually required much concentration.

A list with observed limitation of the above mentioned technique/methods are summarized below as:

S. #.	Author	Technique/ Method Proposed	Limitation
1	Carmel, et al. (2009)	Enhancing Cluster labeling using Wikipedia	➢ Extremely successful in a collection of document whose topics are well covered by Wikipedia concept but the topic that are not covered by Wikipedia may hurt the performance of the proposed technique [9].
2	Treeratpituk, and Callan, (2006),	Automatically labeling Hierarchical Clusters.	➢ It was observed that the performance of proposed technique is affected by clusters containing small numbers of documents.
3	Popsecul, and Unger. (2002)	Automatic Labeling of Document Clusters.	➢ Label selection on the basis of frequent word may not efficiently describe the whole cluster [7]. ➢ None this method gave satisfactory results in the internal node.

4	Tseng, et al. (2006)	Toward Generic Title Generation for Clustered Documents.	➢ Topic that is not covered by database repository may hurt the performance of the proposed technique [7].
5	Ahmed and Khanum, (2010)	"Document Topic Generation in Text Mining by Using Cluster Analysis with EROCK"	➢ Label selection on the basis of frequent word may not efficiently describe the whole cluster [7].
6	Pental K. (2006)	"Automatically Labeling Semantics Classes, and Association for Computational Linguistics"	➢ Label selection on the basis of frequent word may not efficiently describe the whole cluster [7].
7	Maqbool and Babri. (2005)	Interpreting cluster result through cluster labeling	➢ This paper provide a labeling technique for software systems and unable to label text data
8	Mei, et al, (2007)	Automatic Labeling of Multinomial topic models.	➢ This method is based on pure statistical equations. ➢ Label selection on the basis of frequent word may not efficiently describe the whole cluster [7].

Figure 2.4 Existing Techniques

Related to text mining, we are aware of the fact that much of attention have been paid to text clustering rather than cluster labeling. However keeping in view the importance and need of labeling, various area of research has been explored in text labeling. In this section various techniques of cluster labeling are discussed and the common limitations are picked. In order to develop more refine and efficient technique, first of all there is a need to understand the basic concept of cluster labeling, characteristics of a good label and various techniques that are helpful in identifying good labels. All of these key points will be discussed in next chapter.

CHAPTER 3

Cluster Labeling

CLUSTER LABELING

Due to explosive growth in data, there is a need for more advanced tools and techniques that may assists in transforming such data into useful information leading to knowledge. Various classification methods are emerged to group and classify data of same interest. Among all these techniques, the one well known technique is clustering which is used to collect related data into a single point [5]. The process of grouping the data into classes of similar object is called clustering [5]. The data objects inside a cluster should have similar properties with each other, and it should possess that are similar to each other lies within the same cluster, and are dissimilar to data objects belonging to other clusters [1]. We have already discussed clustering techniques briefly in chapter 2. This chapter focused on cluster labeling. Moreover prevailing technique that are in use for cluster labeling are also described in coming sections.

3.1 What is Cluster Labeling?

The simplest definition of cluster labeling is the process of allocating appropriate title to a particular text block/cluster [13]. It may also be called as cluster descriptor [13]. Cluster label efficiently describe the given cluster and how it differs from other clusters [5]. The cluster label may describe a particular cluster in more meaningful way and clearly explains its contents. A cluster label is considered as a good label if it possesses the following, characteristics.

1. It should be of short length and to the point [27].
2. It should be so waightfull that easily convey the topic of the whole cluster [7].
3. It should reflect the central concept of a cluster in such away that user can easily decide that whether a particular cluster is relevant to him or not [27].
4. It should be distinct and will provide distinctiveness among clusters [27].

3.2 The Benefits of Cluster Labeling

When clusters are build up then there is a need to assign appropriate label to each cluster. Cluster labeling is a beneficial task because it helps user to understand the central idea of cluster. Through label user can easily find a particular cluster of his own concern. Cluster labeling resolve the problem of weak understandability and save time of user during cluster analysis [13].

3.3 Characteristics of good Labeling

Some characteristics of a good label are further explained, based on previous research work.

3.3.1 Conciseness: It means that a cluster label should be as short and to the point. It should sufficiently convey the topic of a particular cluster [27]. One way to calculate the conciseness of a label is to measure the length of label where the length is the number of characters or words that exists in a label. To calculate the conciseness of a label a method that uses minimum description length principle and maximum description principle is proposed by Byron & Martin [13].

3.3.2 Comprehensibility: By Comprehensibility we mean the mapping of cluster label to contents of clusters. Simply we can say that comprehensibility is the matching of label with corresponding cluster. This characteristic of cluster labeling is called transparency [13]. It is also referred as the productiveness of labels by Krishna.

3.3.3 Accuracy: Accuracy means that a cluster label should reflect the theme of the particular cluster. "Different from keyword extraction, object of clustering description is not individual document but multiple documents in a cluster with the same object. Tesng & Lin [5] used correlation coefficient and its variation to measure the correlation degree between descriptive word and cluster.

3.3.4 Distinctiveness: Distincti veness describes that cluster label should be most frequent in self cluster and less frequent in other clusters. One well known technique to calculate distinctiveness is represented by Hanan & Mohamed [13]. They proposed a

modified Term Frequency and Inverse Document Frequency (TF and IDF) technique to calculate the uniqueness of a label. Pucktada & Jamie [20] proposed an identical technique to calculate the descriptiveness of cluster descriptor.

3.4 Few valuable technique used in cluster labeling

A few statistical equations that are commonly used by researcher for term extraction and cluster labeling are discussed as under:

TF-IDF: TF-IDF is termed as Term Frequency and Inverse Document Frequency. TF-IDF is a well known statistical technique used in text retrieval [28]. Using this technique, one may find that how much a word is important in a given document. Tf-idf calculates the number of occurrence of any word in its own cluster and also in the rest of document. The weight of word may increase proportionally with the increase in frequency of word in a self cluster, but is decrease by the frequency of the word in other cluster. Tf–idf weighting scheme are widely used in different search engines as a text finding tool and also for ranking a particular document values [28]. Mathematically, in order to compute the weight of the word t_i in a particular document d_j we use the equation [28]:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Where $n_{i,j}$ is the frequency of that particular word (t_i) in given document d_j, and the denominator is the net sum of number of existence of all words of the document d_j, i.e., the size of the document $| d_j |$.

The inverse document frequency (Idf) which is used to calculate the existence of the word/term and can be calculate by dividing the total number of documents by the number of documents containing the corresponding word, and then by applying the log of product.

$$\mathrm{idf}_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

Where $|D|$ is the number of documents in the text and $|\{j: t_i \in d_j\}|$ represent the number of documents having the word t_i.

Finally the TF-IDF is:

$$(\text{tf-idf})_{i,j} = \text{tf}_{i,j} \times \text{idf}_i$$

3.4.1 **CHI-SQUARE TEST:** Chi-Square tests are used to classify data into two or more categories taken from a population. This technique is helpful to check whether there is an important association between the two variables [25]. Chi-Square test is defined as the chi-square is performed to check difference between the expected frequencies and the observed frequencies [25].

Mathematically

$$\chi^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i$$

Where as in the above equation O_i represent observed frequency and E_i is the expected frequency. The value of expected frequency is further calculated by:

$$E_i = N(F(Y_u) - F(Y_l))$$

In above F is the cumulative distribution function for the distribution being tested, Y_u and Y_l is the higher limit and lower limit for class i, and N represents the sample size.

3.4.2 **VECTOR SPACE MODEL:** Vector space is widely used to symbolize text data in the form of vectors. It is an algebraic model which is helpful in information retrieval, document indexing, text word ranking and information filtering [26]. However, the procedure of the vector space model can be divided in to three phases. The initial phase is the term indexing where category specific terms are extracted from the text.

The next phase is the weighting of the indexed terms to strengthen the retrieval user relevant text data. In the last phase documents are ranked with respect to the query by using a similarity measure or similarity coefficient [26].

All of the above statistical techniques are widely used in the field of data mining, pattern extraction and knowledge discovery. The proposed technique is based on term frequency and inverse document frequency discussed in section 3.4.1. The next chapter described the detailed description of the proposed conceptual model.

CHAPTER 4

Proposed Conceptual Model

PROPOSED CONCEPTUAL MODEL

Text mining is basically concern with information extraction, pattern finding and knowledge discovery from large data sets. The process of detecting and deriving valuable and descriptive information from huge text repositories is called text mining [1]. It encapsulates various techniques that handled large databases. Some common and well known techniques are document categorization, document clustering, entity or concept withdrawaling, the production of granular taxonomies, sentiment analysis, text summarization, and entity relation modeling [14]. Because of the shortage of time and space we are unable to discuss all of the above techniques. However in this section we discuss text clustering.

Whenever there is a need to arrange data in groups such that observations possess similar characteristics lies in one group, then we perform clustering. Clustering performs a vital role in decision making, pattern recognition, data mining and knowledge discovery, information retrieval, data partition, image segmentation and machine learning tools [5]. Different clustering algorithms are devised to perform clustering, some common off these are K-mean clustering [5], hierarchical clustering [5], K-mediod [5], Agglomerative clustering [5].

Clustering is used to group unlabeled data. After the grouping of unlabeled data (clusters), we usually interact with unlabeled data (clusters) for searching data of our own interest. There is a need to put suitable label on unlabeled groups or clusters. It may help the user to search data of his interest efficiently. A good label may efficiently describe a particular cluster without going through its all details.

This chapter describes the proposed conceptual model. The diagrammatic representation of the proposed model is shown in Figure 4.1.

The proposed model is consists of two major phases i.e. (i) Pre labeling phase and, (ii) labeling phase. In each phase various steps are performed on given dataset. Below is the detail of each step.

4 Description of Proposed Model:

As we observe in literature review that there exists limitations in existing labeling techniques. In order to overcome these limitations, a technique based on external resources is proposed. The description of each step of the proposed technique is as under:

4.1 Pre-Labeling Phase:

The pre labeling phase may also be called as cluster preprocessing phase in which cluster is formed and then different steps are applied on it to make it clean and eligible for next step. A sequence of steps that may be performed in each phase are discussed in the next section.

4.1.1 Clustering:

Clustering technique is used to breakup text data into semantically related groups [5]. In this section given dataset is partition into smaller, similar and homogeneous groups. As the main focus of this research is on labeling so for clustering we adopt any well known clustering method which combining similar observations. In this research we apply K-means clustering method [5] to cluster our text dataset. The procedure of K-means begins with determining the numbers of clusters. Initially the whole dataset is considered as one cluster with its center is the mean of the dataset. This cluster is break up into two sub parts and the means of the new clusters are iteratively trained [5].These two clusters are again divided into two cluster and the whole procedure is continue until required number of cluster is obtained. The square–error criterion is used, and defined as:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2,$$

Where E is the sum of the errors for all object in given dataset, and p is the point in space. If the specified number of clusters is not a power of two, then the nearest power of two above the number specified is chosen and then the least important clusters are removed and the remaining clusters are again iteratively trained to get the final clusters [5].

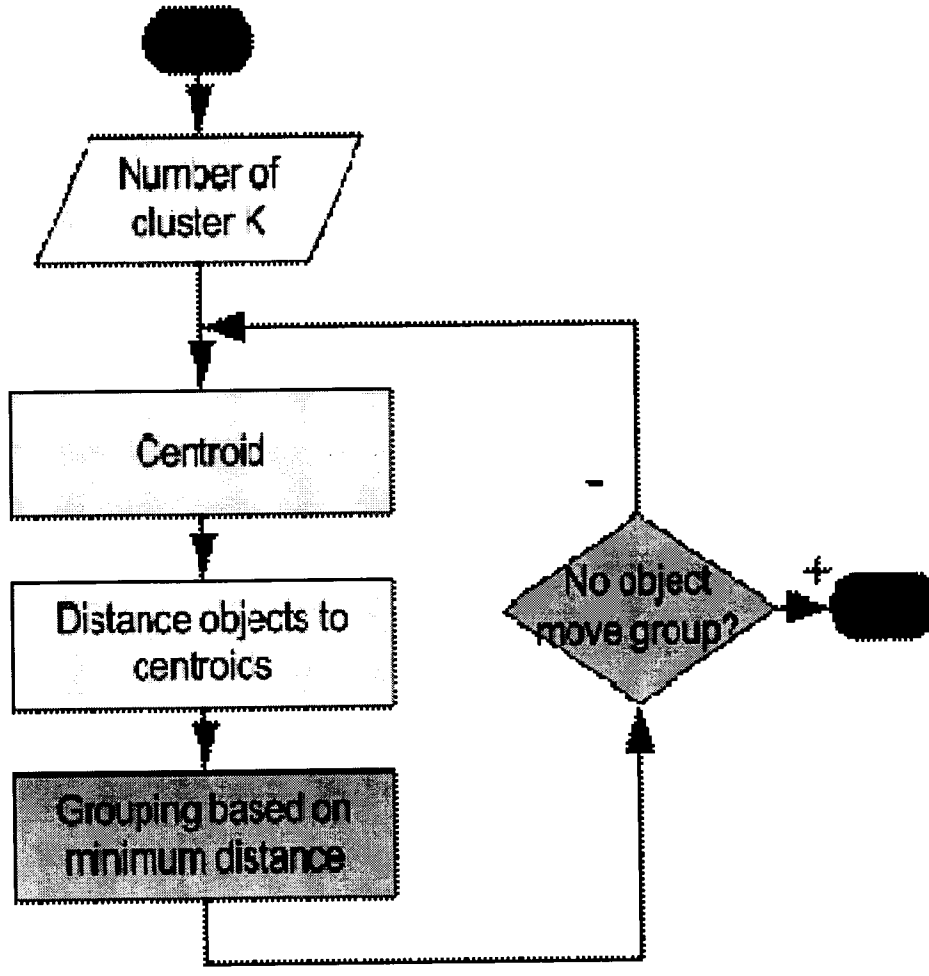


Figure 4.1 K-Mean Clustering

4.1.2 Stemming:

After the formation of clusters the next step is stemming. During clustering process, stemming is also performed, however in some cases stemming is considered as an individual step. The objective of this step is to get rid of unwanted words like commas, articles and full stop. This phase is labeling preprocessing phase. In order to eliminate common word we use standard Porter Stemmer algorithm along with modifications.

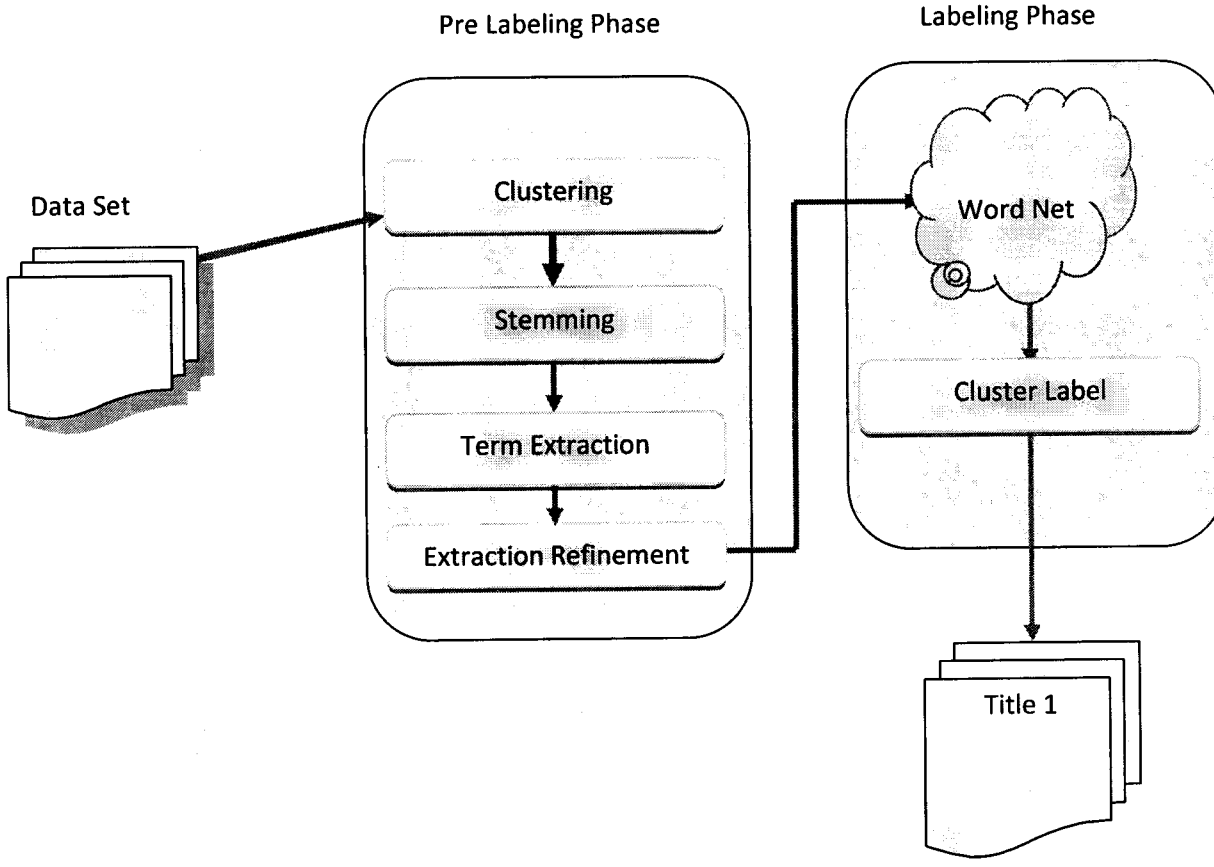


Figure 4.2 Proposed Framework of Cluster Labeling

4.1.3 Term Extraction:

The next step to be performed is term extraction. This step automatically extracts candidate words from each cluster by using term frequency and inverse document frequency technique [28]. The candidate word having higher frequency in self cluster

and lower frequency in the remaining clusters formed so for is selected as candidate word for next step to be performed i.e. extraction refinement.

4.1.4 Extraction Refinement:

The main idea of refinement is taken from the concept of reflective interviewing [33] in which an examiner asks a question from a candidate and then again confirms it from student for personal satisfaction. Same concept is applied here in which each candidate word is further refined by checking the valid thesaurus of it in thesaurus dictionary. To strength the frequency of each candidate word all the observed thesaurus are further checked in particular cluster if they exist, the frequency of that candidate label is incremented, same procedure is applied to all candidate word. After the completion of this phase the word with highest frequency then a common threshold is selected for next phase. The block diagram of this phase is shown as follow:

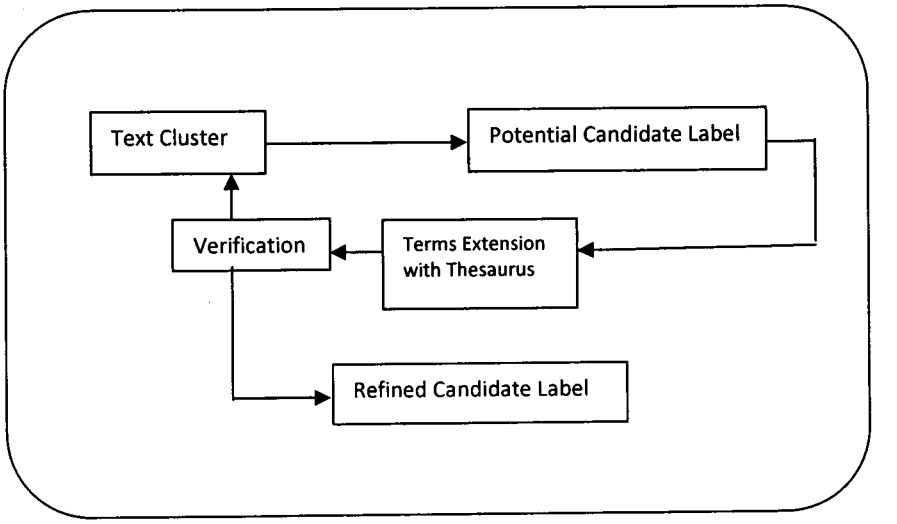


Figure 4.3 An Overview of the Steps of the Pre-Labeling Process.

4.2 Labeling Phase:

After the refinement phase, to choose more suitable potential candidate words for labeling that particular cluster. The next phase which is called Labeling Phase is to be performed, which is the last phase of the proposed technique. This phase generate a

final label for a particular cluster is generate. The detail of each step of this phase is explained as follows:

4.2.1 WordNet:

WordNet is an online hierarchical database system which is considered as a backup for English lexical terms [29]. In WordNet hierarchical structure, words in English language are arranged in a hierarchical manner. Each of the word in WordNet hierarchy has some specific relation to its child and parent word. One common example of such relation is shown below.

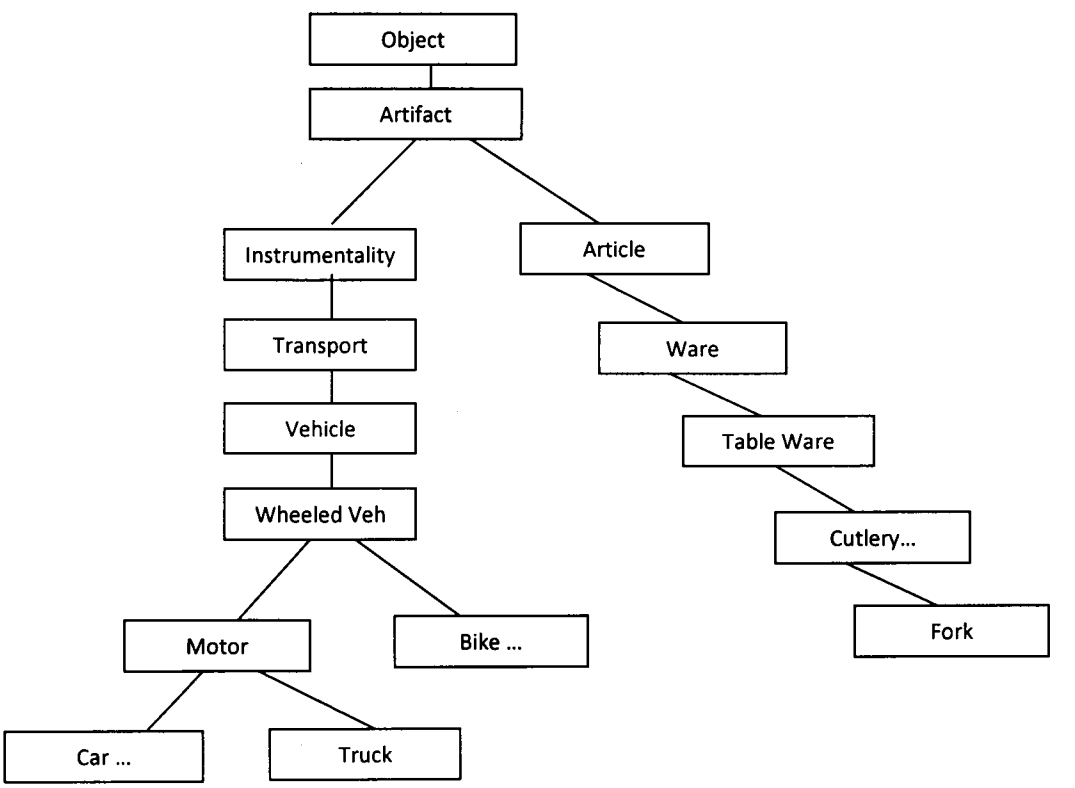


Figure 4.4 WordNet hierarchy adopted from [29].

In the above figure we observe the hyponym structure of WordNet. We also observe the distance length between each word in the hierarchy. The path between each step helps to calculate the relatedness distance between two words.

4.2.2 Final Label Generation:

Final labels are generated from the candidate words by using WordNet. Candidate words extracted is matched with WordNet for identifying a common hypernym. After which the most generic heprnyms among all hypernyms is considered as "label" for that cluster.

4.2.3 Algorithmic Description of Each Phase:

This section described detailed algorithm for each step discussed in the previous section. It explains flow and integration of various steps of the proposed technique.

**/ Clustering..... */*

Step 1. Start with a selection of the number of clusters i.e. Define k = number of clusters

Step 2. Put any early partition that divide the given data into k clusters. The training sample may be assign systematically or randomly as described below:

1. Take the first k object as single cluster.
2. Assign each of the remaining (N-k) objects to the cluster with the closest centroid. After first iteration, recomputed the centroid.

Step 3. Take each object in proper sequence and calculate its distance from the centroid of each cluster. If an object does not exist in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4. Repeat step 3 until all samples are clustered, that is until a pass through the training sample causes no new assignments.

NOTE: In case if the size of data set is less than the number of cluster then allocate each data object as the centroid of the cluster. And consider Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, we calculate the distance to all centroid and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data.

**/ Stemming..... */*

Step 1. Input unstemmed text files.

Step 2. Matched each word with common word list

Step 3. If match found then

eliminate that word from wordlist

else

move to next word

Step 4. Repeat step #3 until all words are scan.

Step 5. Obtain a final modified text file.

Step 6. Exit.

Term Frequency Algorithm...../*

Step 1. Input text cluster

Step 2. Initialize an array of n element containing each unique word of cluster with count = 1

Step 3. If unique word encounter / reappear then

increment count as count = count + 1

else

move to next word

Step 4. Repeat step #3 until all frequent words are scan or encounter.

Step 5. Obtain a final array of words with its frequency.

Step 6. Exit.

Extraction Refinement Algorithm...../*

Step 1. Input an array of candidate words with its frequency (continue from step 5 of previous algorithm)

Step 3. Obtain valid thesaurus of each candidate word from database repository.

Step 2. Match each valid thesaurus with word's corresponding cluster.

Step 3. If match found then for each candidate word

Increment count as count = count + 1

else

move to next word (thesaurus).

Step 4. Repeat steps #3 until all valid thesauruses are scan or encounter.

Step 5. Obtain a final and incremented array of candidate words with its frequency.

Step 6. Exit.

Final label generation Algorithm...../*

Step 1. Input an array of potential candidate words after applying a common threshold.

Step 2. Match each potential candidate word with WordNet.

Step 3. If match found then

Select that matched hypernym of word net

Else

Mark the potential candidate word.

Step 4. Repeat step #4 until a possible hypernym is obtained for all potential candidate words.

Step 5. Choose the most frequent hypernym as a label of the corresponding cluster

Step 6. Exit.

This section describes the complete steps along with its algorithms of the proposed technique. In next chapter we discussed the experimental setup based on the above algorithms. The results so for obtained by executing the proposed technique will be compare with existing technique.

CHAPTER 5

Implementation and Result

IMPLEMENTATION AND RESULT

In previous chapter we discussed about the proposed technique and the algorithm designed for cluster labeling. In this chapter we give the detailed description of the experimental setup and implementation of the proposed technique. After applying the proposed techniques, results obtained are also compared with different techniques already developed and used by other researchers.

5.1 Experimental Environment

The technique proposed in our research has been implemented in C#. The code is presented in appendix A. The snapshot of developed technique inside a visual programming is shown as under:

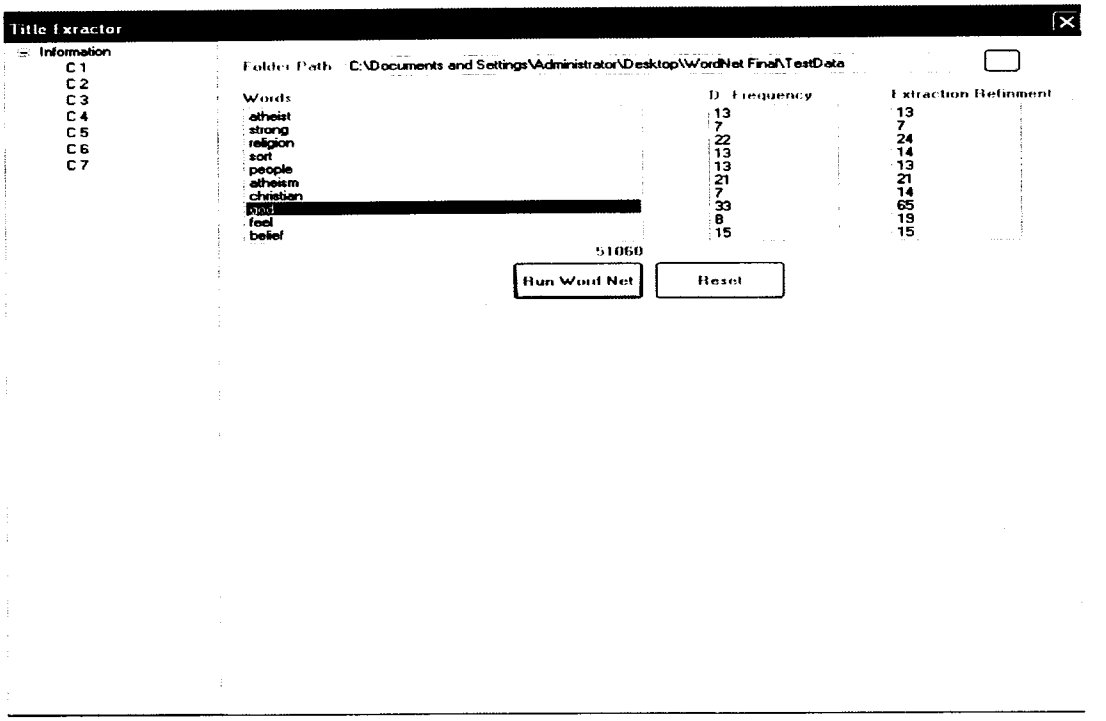


Figure: 5.1 View of proposed Technique

5.2 Dataset for Experiment

Three different types of text datasets were selected to carry out experimental work. All of these datasets were rich text material. The first data set is taken from the "Daily Jang" newspaper. The whole document is considered as one cluster containing information about different types of games. The document consists of 2000 words.

Second dataset was a collection of Reuters-21578. The dataset is available at http://www.daviddlewis.com/resources/testcollections/reuters21578/. It has also been frequently used for experiments by different researchers in text mining. Reuters-21578 is the most commonly used dataset for text mining and text categorization. The documents inside the given dataset were then transformed into cluster using k-mean clustering [5] technique. Since this research is based on cluster labeling so clustering is usually assumed or performed by using common clustering algorithms.

The third dataset was "20 Newsgroups" dataset is a group of almost 20,000 web messages, these messages are further divided into 20 different groups. 20 newsgroups is well accepted dataset for experiments in text applications of machine learning techniques, such as text classification and text clustering [9].

5.3 Experimental Results

This section present the detailed description of the experiment performed so for. Detail of each steps performed and its experimental result are tabulates in the next coming section.

(i) Experiment 1:

The first step was to apply stemmer algorithm on this dataset. The contents of first dataset are reduced up to 65% after applying modified stemmer algorithm. This algorithm eliminates postfix and common words like full stop, commas and words having frequency less than or equal to 3. The next step was term extraction, which was carried out using term frequency and inverse document frequency technique. It resulted in extracting the most frequent words with higher frequency occurrence which was about 7% of the dataset. The results were further refined in the next step using thesaurus of the selected words and searching of their existence in the dataset leading to updating frequency of the corresponding words selected in previous step. It reduced

the occurrence of frequent words upto 5 % and was considered potential candidate words for labeling the dataset. The result of each step is shown in the below table 5.1.

Table: 5.1 *A detail description of the steps performed before labeling.*

Cluster	***Most Frequent word along with its frequency***		***Extended frequency after refinement***	
	Top Word	***Frequency***	***Top Word***	***Frequency***
SPORT	*1. Football* *2. Player* *3. Hockey* *4. Ground* *5. Cricket* *6. Captain* *7. Match*	*68* *55* *39* *38* *37* *20* *19*	*1. Football* *2. Player* *3. Hockey* *4. Cricket* *5. Match*	*68* *59* *39* *37* *33*

The selected top words were mapped with WordNet to generate final labels for the dataset under consideration. The final labels generated are mentioned in table 2. The labels generated truly matches with the label manually assigned to the document, when it was compare with it. In this regard maximum accuracy has been achieved leading to nearly 100%. Results so for obtained is presented in the below table. 5.2.

Table 5.2: *A detail description of the steps performed after labeling.*

Cluster	***Top Words Selected***	***Top Words Selected after Refinement***	***Final Label Generated through Proposed Technique***
SPORT	*1. Football* *2. Player* *3. Hockey* *4. Ground* *5. Cricket* *6. Captain* *7. Match*	*Football* *Player* *Hockey* *Cricket* *Match*	*1. A type of* ***sport.*** *2.* ***Sports*** *Man.* *3. A type of* ***sport.*** *4. A type of* ***sport.***

The comparison of label generated using proposed technique and label generated manually reflects almost 100% similarity. The label obtained by applying proposed technique is reflects the subject/theme of the cluster accurately. The graphical

representation (see figure 5.3) shows a comparison of results for the accuracy achieved of proposed technique for labeling and manual labeling technique.

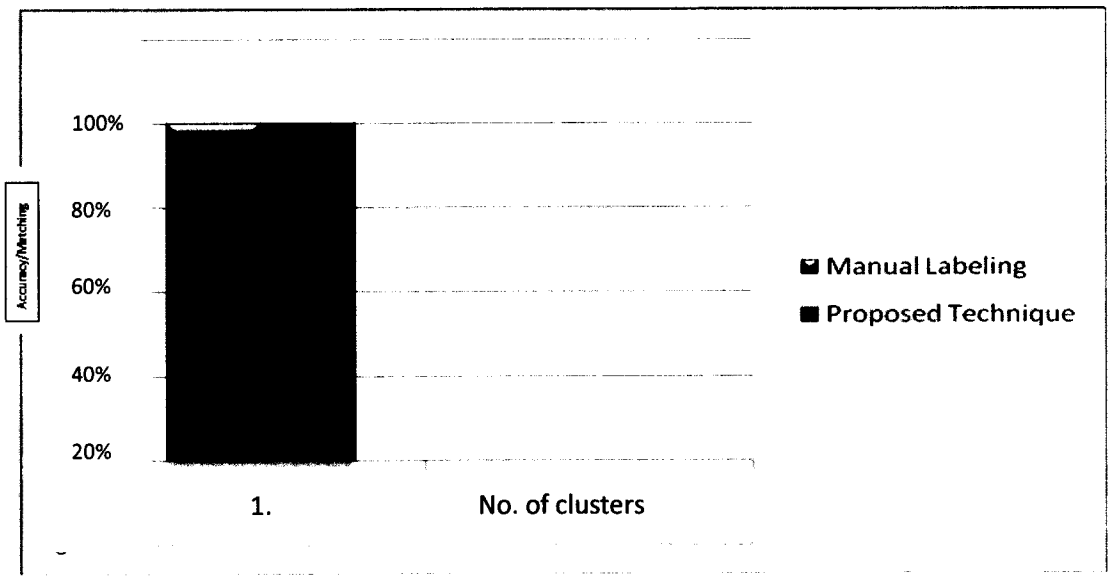


Figure: 5.3 Comparison of Propose technique with Manual Labeling

(ii) Experiment 2:

The second collection was gathered by downloading pages from the Open Directory Project (ODP) [14]. For this purpose, we randomly selected 100 different categories from the ODP hierarchy. Example categories include, among others, sub-categories of the top level ODP categories such as Ceramic Art and Pottery. From 143 each category we then randomly selected up to 100 documents, resulting in a collection size of about 10,000 documents. In both collections, the categories were manually labeled. These ground-truth "correct" labels were later used to evaluate our labeling system.

(ii) Experiment 3:

Since the reuter-21578 is almost 2MB data arrange in different classes. Among all these 06 largest categories that contain almost 6000 stories were choose for an experiment. In the first step the basic term frequency and inverse document frequency technique is used to pick 7% of potential candidate words. The results were further

ID	Category/ Cluster	Top Words	Refined Word	Final Label against each word
1	Animals	Rabbit, John, Horse, Cluster, Name, Dog	Rabbit, Horse, Cluster, Dog	1. Herbivorous/**Animal** 2. Herbivorous/ **Animal** 3. Group of similar things 4. Carnivores/ **Animal**
2	Automobile Information	CNG, Fuel, Truck, Car, Automobile, Road	CNG, Fuel, Truck, Car	1. A substance 3. A **vehicle** 4. A **vehicle**
3	Air Line Information	Column, Scan, New, John, Code	Code, New	1. Unfamiliar, Unknown **2. Rules, Principle, Law**
4	Language	John, Claim, Enough, Cluster, Germany	Claim, Enough, Cluster	1. Demand for something 2. Sufficient for something **3. Grouping of similar thing.**
5	Male Expectation	Life, Age, Year, Africa, Expectation	Life, Age, Year, Expectation	1. Mode of Living 2. How long something exists. 3. Period of time **4. Expectation**
6	Protein Amount	Fat, Protein, Beef, Amount, Calcium	Fat, Protein, Beef, Calcium	1. Bodily Property 2. Substance of Egg 3. Beef Cattle 4. Metallic Item

(ii) Experiment 3:

Since the reuter-21578 is almost 2MB data arrange in different classes. Among all these 06 largest categories that contain almost 6000 stories were choose for an experiment. In the first step the basic term frequency and inverse document frequency technique is used to pick 7% of potential candidate words. The results were further

By comparing all the candidate words with WordNet it was clearly observed that 85% accuracy is achieved. Results so for obtained are shown in figure 5.6. And the comparison of the proposed technique with manual labeling technique and already existing technique is shown in below table 5.5.

Table 5.5: *A detail description of the steps performed after labeling.*

Manual Label	Top Words	Refine words	Final Labe generated by proposed technique
Earn	1. NET 2. QTR 3. Shr 4. Cts 5. Net 6. Revs	1. NET 2. QTR 3. Net 4. Revs	1:goal 2:trap 3:**income** 4: **income**
Acquire	1. Acquire 2. Acquisition 3. Stake 4. Company 5. Share	1. Acquire 2. Stake 3. Share	1:device 2:stock certificate, stock 3:wedge
Money	1. Currency 2. Money 3. Market 4. Central banks 5. The Bank 6. Yen	1. Currency 2. Money 3. Market 4. Yen	1:**Currency** 2:**Currency** 3:marketplace, mart 4: China **Currency**
Grain	1. Wheat 2. Grain 3. Tones 4. Agriculture 5. Corn	1. Wheat 2. Grain 3. Corn	1:seed/ eating **food** 2:cereal, cereal grass 3:**foodstuff**, 4:**food** product
Crude/fuel	1. Crude oil 2. bpd 3. OPEC, 4. mln barrels 5. Petroleum	1. Crude oil 2. OPEC, 3. Petroleum	1:lipid, lipide, lipoid 2:**fuel/oil** 3:**fossil fuel**
Trade	1. Trade 2. Tariffs 3. Trading 4. Surplus 5. Deficit 6. Gatt	1. Trade 2. Tariffs 3. Trading 4. Surplus	1:**business** 2:UN **business** agency 3:prevailing wind 4: **Business** rule

Crude/fuel	1. Crude oil 2. bpd 3. OPEC, 4. mln barrels 5. Petroleum	1. Crude oil 2. OPEC, 3. Petroleum	1:lipid, lipide, lipoid 2:**fuel/oil** 3:**fossil fuel**
Trade	1. Trade 2. Tariffs 3. Trading 4. Surplus 5. Deficit 6. Gatt	1. Trade 2. Tariffs 3. Trading 4. Surplus	1:**business** 2:UN **business** agency 3:prevailing wind 4: **Business** rule

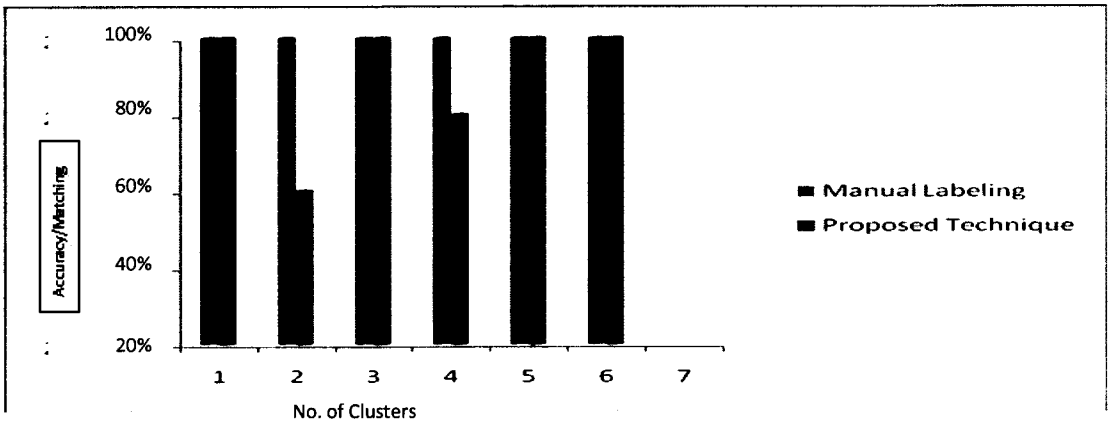


Figure: 5.6 Comparison of Propose technique with Manual Labeling

Results are compared with two alternative techniques. The manual labels of the reuters-21578 assigned by humans and label generated using technique proposed by Tseng [7], are shown in table 5.7.

Manual Label	Tseng Technique	Final Label using WordNet
Earn	1:trap:1.000 2:game equipment:1.000 3:fabric, cloth, textile:1.000	1:goal 2:trap 3:**income** 4: **income**
Acquire	1:asset:0.500	1:device 2:stock certificate, stock 3:wedge
Money	1:medium of exchange, **monetary system**:0.750	1:**Currency** 2:**Currency** 3:marketplace, mart 4: China **Currency**

Grain	1:weight unit:1.250 2:grain, **food grain**:1.250 3:cereal, cereal grass:1.250	1:seed/ eating **food** 2:cereal, cereal grass 3:**foodstuff**, 4:**food** product
Crude/fuel	1:oil:0.750 2:**fossil fuel**:0.750 3:lipid, lipide, lipoid:0.500	1:lipid, lipide, lipoid 2:**fuel/oil** 3:**fossil fuel**
Trade	1:liability, financial obligation, indebtedness, pecuniary obligation:0.062	1:**business** 2:UN **business** agency 3:prevailing wind 4: **Business** rule

Table: 5.7 *Comparison of Proposed Technique with Manual Labels and existing Technique [7].*

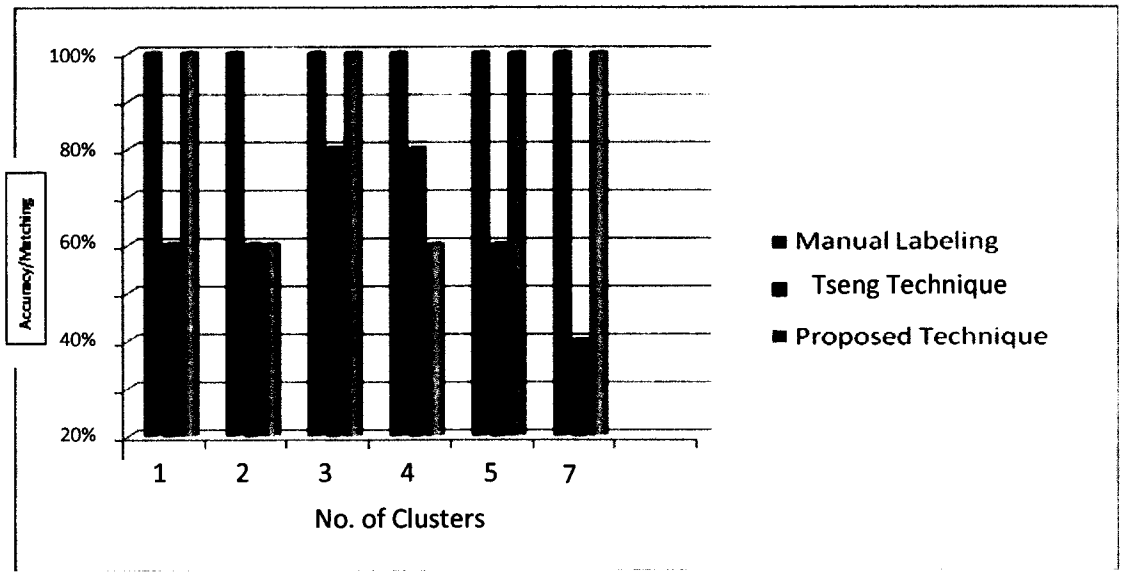


Figure: 5.8 Comparison of Propose technique with Manual Labeling and existing technique

After comparing results with the existing techniques, we came to know that the proposed technique shows 75% accuracy and the result obtained are almost same as generated by manual labeling after wasting large time. The result observed by our proposed system is also compare with the result obtained by term frequency method.

(5.4) Discussion:

The results obtained by applying the proposed technique on reuter-21578 text data set reflects the performance of the proposed technique is outstanding in term of accuracy can not be underestimated. The main constraint that was observed in previous techniques in use for labeling is that the most of them used only term frequency and

upon statistical equations. Thirdly, there exist many manual labeling techniques although they are justifiable but they are costly and time consuming [15].

After the comparison of proposed technique with other existing techniques it is clear that the performance of proposed technique is outstanding. However, the technique proposed also have some constrains. WordNet doesn't cover all the terms extracted from text clusters. Also some of the WordNet generated titles may not reflect the theme of a particular cluster accurately. If the text inside a cluster bitterly matches with the hierarchy of WordNet, its results in improved performances. One common issue relating to the achievement of high quality results is the process of clustering performed before applying the existing technique. If a suitable clustering is found in a dataset then the proposed technique may lead to more desirable results. After comparing results with the proposed technique it was observed that almost 60% of shortcoming discussed at page number 18 has been addressed.

There is no boundary of research and there always exist a gap for new effort. Cluster labeling using WordNet and thesaurus is novel attempt that may produce better outcome. The graphical view results mention at page number 37-41 reflects that almost seventy percent labels correctly matches with manual labels. Almost all of the generated labels accurately reflect the theme of that cluster used in experiments. The limitation of WordNet may impede the accur acy after label generated. The next chapter addressed the possible suggestions relating to proposed technique and how it can be further improved. It also addressed the efforts so for taken in order to develop the proposed technique and further extension.

CHAPTER 6

Conclusion and Future Work

CONCLUSION AND FUTURE WORK

Cluster labeling is the process of allocating appropriate title to a particular cluster. Tremendous research work in the field of clustering and text grouping has been observed in the past research so for, however research relating to cluster labeling and text document title generation invites researchers attention to explore new or extend its existing techniques. Regarding to this various techniques is highlighted and discuss in Chapter No 2. Although all of these techniques are valid and justifiable, but there still exist various limitation in these techniques. Some of these limitations are given as under:

- Most common shortcoming that arise in manual labeling technique i.e. they are costly and time consuming.

- Various techniques in practice for text labeling select the label which is not almost generic. The labels selected are often part of the document itself which may not reflect the theme strongly especially when the document encompasses wide range of related/relevant aspects.

- Particularly for text clustering and labeling the statistical equations used for labeling a cluster, provide results (i.e. label of cluster) that may not be considered as a good cluster descriptor.

In order to eliminate such common shortcomings, this research effort presents a technique to label a cluster using an external resource such as WordNet and thesaurus. Candidate words are selected by using different steps like term frequency, stemming and thesaurus. After selecting candidate words the selected words are matched with the WordNet to obtain the final label.

Our research attempted to overcome the shortcomings appeared in the past research by introducing an extended new tools and techniques for labeling a cluster such that a label may better interpret the cluster on its part. The propose technique is based on the basic concept of clustering as well as labeling, and how it should be improved by using external recourse WordNet. Instead of selecting labels from the text cluster itself the proposed technique produces a generic label that efficiently describe, encompassing the theme of the cluster, to be labeled. This labeling technique automatically labels the cluster in an efficient way. The usefulness of WordNet and thesaurus is also apparent in labeling a cluster.

6.1 Future Out Look

There exists always gap what we know and what we intend to know. The research end over provides new ways to thing process leading to the new idea to be generated and tested to explore the knowledge. Cluster labeling using WordNet and thesaurus is an attempt that produces better result as compared to prevailing technique in use.

Our research work has laid a base to further extend the existing labeling technique based on the idea to use external resources for text labeling purposes. It gives the future direction to us and to define more refined structure on the basis of external resources. Our proposed technique provides an easy way to chook a label for a cluster. Cluster labeling by using WordNet and thesaurus is extremely successful, as shown by our results. The label generated truly matches the label already given to the document used as dataset. Its accuracy is almost 80%.

After examine the performance of existing technique it was observed that few labels were not covered properly because of its less importance in WordNet. Moreover we observe that some topic require multi topic label. The proposed framework may be unable to allocate multi topic label to a particular document. And there exists is a need to take an intelligent decision regarding to multi topic labeling. This will be consider as future work or enhancement to the propose technique.

Appendix A

C# Code of Proposed Model

SIMPLE C# CODE FOR AN INTEGRATED APPROACH TO LABEL CLUSTERS

This is the code which actually read files from selected folder, merge its text into one file. Remove all the noise words from that doc. Make tree from that merged files, after select each one will parse that file for mostly repeated (high frequency) top 10 words. Then find synonyms of each word and find its frequency, add to its source word. At last when applied the hypernym over the top five frequent words we will get the suitable name of that cluster.

```
**********************************************************************************************
*********************************************
        *******************************************

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Collections;
using System.Xml;
using System.Text.RegularExpressions;
using System.Diagnostics;
using mshtml;


namespace WordNet.App
{
    public partial class TitleExtrator : Form
    {
        Hashtable UniqueWords;
        IHTMLDocument2 htmlDocument;
        IHTMLDocument2 docPara = null;
        ArrayList stopWords = null;
        public TitleExtrator()
        {
            InitializeComponent();
            stopWords = new ArrayList();
            TextReader tr = new StreamReader("stopwords.txt");
            string singleWord = tr.ReadLine();
            while (singleWord != null)
            {
                stopWords.Add(singleWord);
                singleWord = tr.ReadLine();
            }
            tr.Close();
            this.webBrowser1.DocumentText = "";
            docPara = this.webBrowser1.Document.DomDocument as IHTMLDocument2;
            docPara.designMode = "On";


        }


        private void button1_Click(object sender, EventArgs e)
        {
```

```
            FolderBrowserDialog folderDlg = new FolderBrowserDialog();
            ArrayList UnwantedWords = Stemmer.UnwantedWords();
            if (folderDlg.ShowDialog() == DialogResult.OK)
            {
                this.txtFolder.Text = folderDlg.SelectedPath;
                TreeNode ParentNode = new TreeNode("Information");

                string[] files = Directory.GetFiles(folderDlg.SelectedPath);
                for (int i = 0; i < files.Length;i++ )
                {
                    TreeNode tempNode = new TreeNode("C " + (i + 1));
                    tempNode.Name = files[i];
                    string text = "";
                    TextReader txtReader = new StreamReader(files[i]);
                    text = txtReader.ReadToEnd();
                    txtReader.Close();
                    //=================================================================
                    try
                    {

                        //Counting Words and Save it for maximum occurance
                        UniqueWords = new Hashtable();
                        string[] words = text.Split(new char[]{' ','\n'});
                        for (int r=0;r<words.Length;r++)
                        {
                            string word = words[r];
                            word = word.Replace("\r\n", "");
                            word = word.EndsWith("ies") ? word.Replace("ies", "y") :
word;
                            char[]
endschar={',','.','?','\"','\'',':','\\','%','#','!','(',')','-','_','=','+','$'};
                            word = word.TrimEnd(endschar);
                            word = word.TrimEnd(new char[]{'s'});
                            word = word.TrimStart(endschar);
                            word = Regex.Replace(word, "[0-9.]", "");
                            word = word.EndsWith("ed") ? word.Replace("ed", "") : word;
                            word = word.EndsWith("ing") ? word.Replace("ing", "") :
word;
                            if (word != "")
                            {
                                string tempWord = word.Replace("(", "").Replace(")",
"").Replace("*", "").Replace("[", "").Replace("]", "").Replace("?", "");
                                text = Regex.Replace(text, tempWord, word);
                            }
                            text = text.Replace("\"","");

                            if (word.Length > 2 &&
!UnwantedWords.Contains(word.ToLower()))
                            {
                                word = word.Replace("(", "").Replace(")",
"").Replace("*", "").Replace("[", "").Replace("]", "").Replace("?", "");
                                int count = Regex.Matches(text, " "+word +"
",RegexOptions.IgnoreCase).Count;
                                if (!UniqueWords.Contains(word.ToLower()) &&
!stopWords.Contains(word.ToLower()))
                                {
                                    UniqueWords.Add(word.ToLower(), count);
                                }
                            }
                            word = "";
                        }

                        //Sorting the Unique words occurance
                        Dictionary<string, int> dict = new Dictionary<string, int>();
                        foreach (DictionaryEntry st in UniqueWords)
                        {
                            dict.Add(st.Key.ToString(), int.Parse(st.Value.ToString()));
                        }
                        List<KeyValuePair<string, int>> summaryList = new
List<KeyValuePair<string, int>>();
                        summaryList.AddRange(dict);
```

```
                        summaryList.Sort
                        (
                            delegate(KeyValuePair<string, int> kvp1,
                            KeyValuePair<string, int> kvp2)
                            {
                                return Comparer<int>.Default.Compare(kvp1.Value,
kvp2.Value);
                            }

                        );

                        //Getting the top five words
                        UniqueWords.Clear();
                        if (summaryList.Count > 10)
                        {
                            for (int j = summaryList.Count - 1; j > summaryList.Count -
11; j--)
                            {
                                UniqueWords.Add(summaryList[j].Key,
summaryList[j].Value);
                            }
                        }
                        else
                        {
                            for (int j = 0; j < summaryList.Count; j++)
                            {
                                UniqueWords.Add(summaryList[j].Key,
summaryList[j].Value);
                            }
                        }
                        summaryList.Clear();
                        tempNode.Tag = UniqueWords;
                        tempNode.Name = Path.GetFileNameWithoutExtension(files[i]);
                        ParentNode.Nodes.Add(tempNode);

                    }
                    catch (IOException)
                    {
                        MessageBox.Show("error reading " + files[i]);
                        break;
                    }
                }
                if (ParentNode.Nodes.Count > 0)
                {
                    treeView1.Nodes.Add(ParentNode);
                    treeView1.ExpandAll();
                }
                else
                {
                    MessageBox.Show("No Informative file found in the selected
folder","Information",MessageBoxButtons.OK,MessageBoxIcon.Information);
                }
            }
        }

        private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)
        {
            lstWords.Items.Clear();
            this.lstIDF.Items.Clear();
            lstOccurances.Items.Clear();
            if (treeView1.SelectedNode.Tag != null)
            {
                this.Cursor = Cursors.WaitCursor;
                Hashtable hash = treeView1.SelectedNode.Tag as Hashtable;
                foreach (DictionaryEntry entry in hash)
                {
                    lstWords.Items.Add(entry.Key.ToString());
                    MainForm objMainForm = new MainForm();
                    MainForm.execType = "short";
                    objMainForm.mWordTextBox.Text = entry.Key.ToString();
                    objMainForm.WordDefinition();
```

```
                    int synonymCount = 1;
                    foreach (string synonym in MainForm.synonymsList)
                    {
                        synonymCount = synonymCount + SynonymCount(synonym);
                    }
                    lstOccurances.Items.Add((entry.Value).ToString() == "0" ? "1" :
(entry.Value).ToString());
                    lstIDF.Items.Add(synonymCount+int.Parse(entry.Value.ToString()));
                    objMainForm.Dispose();
                }
                this.Cursor = Cursors.Default;
                lblFileName.Text = treeView1.SelectedNode.Name;
            }
        }
        public int SynonymCount(string search)
        {
            int count = 0;
            string[] files = Directory.GetFiles(this.txtFolder.Text.Trim());
            for (int i = 0; i < files.Length; i++)
            {
                string[] filePath = files[i].Split(new char[] { '\\' });
                string[] fileName = filePath[filePath.Length - 1].Split(new char[] { '.'
});
                if (treeView1.SelectedNode.Name == fileName[0])
                {
                    TextReader tr = new StreamReader(files[i]);
                    try
                    {
                        string FileText = tr.ReadToEnd();
                        int res = Regex.Matches(FileText, " " + search + " ",
RegexOptions.IgnoreCase).Count;
                        count = count + res;
                        tr.Close();
                    }
                    finally
                    {
                        tr.Close();
                    }
                    break;
                }
            }
            return count;
        }
        private void btnReset_Click(object sender, EventArgs e)
        {
            treeView1.Nodes.Clear();
            lstWords.Items.Clear();
            lstOccurances.Items.Clear();
            txtFolder.Clear();
            btnFolder.Focus();
            btnWordNet.Enabled = false;
            this.pnlWordNet.Controls.Clear();
            this.lstIDF.Items.Clear();
        }

        private void btnWordNet_Click(object sender, EventArgs e)
        {
            try
            {
                this.pnlBasic.Visible = false;
                this.pnlWordNet.Controls.Clear();
                string currentDir = Directory.GetCurrentDirectory() + "\\Dict2\\";
                this.Cursor = Cursors.WaitCursor;
                if (lstWords.SelectedItem == null)
                {
                    MessageBox.Show("Please select any word from the list",
"Information", MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
                else if (File.Exists(currentDir +
lstWords.SelectedItem.ToString()+".htm"))
                {
```

```
                    this.pnlBasic.Visible = true;
                    this.pnlWordNet.Controls.Add(pnlBasic);
                    TextReader htmReader=new StreamReader(currentDir +
lstWords.SelectedItem.ToString() + ".htm");
                    string htmlDocument=htmReader.ReadToEnd();
                    htmReader.Close();
                    this.webBrowser1.Document.Body.InnerHtml = htmlDocument;
                    string temp1 = Regex.Match(htmlDocument, "is a kind of.*?[\r|\n|
]*<A.*?<BR>",RegexOptions.Singleline|RegexOptions.IgnoreCase).Value;
                    string temp2 = temp1.Replace("is a kind of", "");
                    string
temp3=Regex.Replace(temp2,"</?a.*?>","",RegexOptions.IgnoreCase|RegexOptions.Singleline)
;
                    string temp4 = Regex.Replace(temp3, "</?span.*?>", "",
RegexOptions.IgnoreCase | RegexOptions.Singleline);
                    string temp5 = Regex.Replace(temp4, "<br.*?>", "",
RegexOptions.IgnoreCase | RegexOptions.Singleline);
                    temp5 = temp5.Replace("\r", "").Replace("\n", "");
                    temp5 = Regex.Replace(temp5, "[ ][ ]+", " ");
                    string[] parents = temp5.Split(new char[] { ',',';' });
                    lstTitle.Items.Clear();
                    foreach (string st in parents)
                    {
                        lstTitle.Items.Add(st.Trim());
                    }
                }
                else
                {
                    MainForm.execType = "";
                    MainForm objMainForm = new MainForm();
                    objMainForm.ClearWordHitory();
                    objMainForm.mWordTextBox.Text =
this.lstWords.SelectedItem.ToString();
                    objMainForm.WordDefinition();
                    //MainForm.synonymsList;
                    objMainForm.FormBorderStyle = FormBorderStyle.None;
                    this.pnlWordNet.Controls.Clear();
                    this.pnlWordNet.Controls.Add(objMainForm);
                    objMainForm.Show();
                }
            }
            finally
            {
                this.Cursor = Cursors.Default;
            }
        }

        private void lstWords_Click(object sender, EventArgs e)
        {
            this.btnWordNet.Enabled = true;
        }

        private void TitleExtrator_Load(object sender, EventArgs e)
        {

        }

    }
}
```

```
This is a test form come with wordnet API, which has functions for defining a word, its
synonyms, detail in web browser.
*****************************************************************************************
*****************************************
        ******************************************


using System;
using System.Linq;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using WordNet.Common;
using WordNet.App.Properties;
using System.Collections;

namespace WordNet.App
{
    public partial class MainForm : Form
    {
        public static ArrayList synonymsList = null;
        public static string execType="";

        #region Constructors
        public MainForm()
        {
            InitializeComponent();
            this.TopLevel = false;
            RebindWordHistory();
        }
        #endregion Constructors

        #region Methods DisplayResultsShort
        private void DisplayResultsShort(string orgWord, Dictionary<string,
List<Definition>> results)
        {
            StringBuilder sb = new StringBuilder();
            synonymsList = new ArrayList();
            if (results.Count > 0)
            {
                #region Prep for sorting, build and rendering
                List<string> words = new List<string>();
                Dictionary<string, List<Definition>> defSets = new Dictionary<string,
List<Definition>>();
                #endregion

                #region Sort results by part of speech
                foreach (string key in results.Keys)
                {
                    foreach (Definition def in results[key])
                    {
                        string pos = def.DisplayPartOfSpeech;
                        if (!defSets.ContainsKey(pos))
                            defSets.Add(pos, new List<Definition>());

                        defSets[pos].Add(def);
                        foreach (string word in def.Words)
                        {
                            if (!words.Contains(word))
                            {
                                words.Add(word);
                                if (word != orgWord)
                                {
                                    synonymsList.Add(word);
                                }
                            }
                        }
                    }
```

```
                }
                #endregion;
            }
        }
        #endregion

        #region DisplayResults
        private void DisplayResults(string orgWord, Dictionary<string, List<Definition>>
results)
        {
            mWebBrowser.Navigate("about:blank");
            HtmlDocument resultDoc = mWebBrowser.Document;
            StringBuilder sb = new StringBuilder();
             if (results.Count > 0)
            {
                #region Prep for sorting, build and rendering
                List<string> words = new List<string>();
                Dictionary<string, List<Definition>> defSets = new Dictionary<string,
List<Definition>>();
                #endregion

                #region Sort results by part of speech
                foreach (string key in results.Keys)
                {
                    foreach (Definition def in results[key])
                    {
                        string pos = def.DisplayPartOfSpeech;
                        if (!defSets.ContainsKey(pos))
                            defSets.Add(pos, new List<Definition>());

                        defSets[pos].Add(def);
                        foreach (string word in def.Words)
                        {
                            if (!words.Contains(word))
                            {
                                words.Add(word);
                            }
                        }
                    }
                }

                #endregion

                #region Build markup for browser control
                foreach (string key in defSets.Keys)
                {
                    StringBuilder defText = new StringBuilder("<ul>");
                    foreach (Definition def in defSets[key])
                    {
                        string formattedDefinition =
FormatDefinition(def.DefinitionText);
                        if (!string.IsNullOrEmpty(formattedDefinition))
                        {
                            defText.AppendLine(string.Format("<li>{0}</li>",
formattedDefinition));
                        }
                    }
                    defText.AppendLine("</ul>");
                    sb.AppendFormat(Resources.PartOfSpechFormat, string.Format("{0}
<sup>{1}</sup>", orgWord, key), defText.ToString());
                }

                string[] wordLinks = (from word in words orderby word ascending select
string.Format(Resources.LinkedWordFormat, word)).ToArray();
                sb.Append(string.Join(", ", wordLinks));
                #endregion
            }
            else
            {
                #region Build no results markup
```

```
                sb.AppendFormat("<h1>No match was found for \"{0}\".</h1><br />Try your
search on <a href=\"http://wordnet.princeton.edu/perl/webwn?s={0}\"
target=\"_blank\">WordNet Online</a>", orgWord);
                #endregion
            }

            #region Write markup to browser doc and refresh
            resultDoc.Write(string.Format(Resources.ResultPageFormat, sb.ToString()));
            mWebBrowser.Refresh();
            #endregion
        }
        #endregion DisplayResults

        #region FormatDefinition
        private string FormatDefinition(string text)
        {
            string retVal = string.Empty;
            if (!string.IsNullOrEmpty(text))
            {
                int exStart = text.IndexOf('"');
                if (exStart > -1)
                {
                    retVal += "<strong>";
                    retVal += text.Insert(exStart, "</strong><br /><i>");
                    retVal += "</i>";
                }
                else
                {
                    retVal = string.Format("<strong>{0}</strong>", text);
                }
            }
            return retVal;
        }
        #endregion FormatDefinition

        #region DefineWord
        private void DefineWord(string word)
        {
            //AddWordToHistoryBar("0");
            Dictionary<string, List<Definition>> definitions =
DictionaryHelper.GetDefinition(word);

            if (MainForm.execType == "short")
            {
                DisplayResultsShort(word, definitions);
            }
            else
            {
                DisplayResults(word, definitions);
            }
        }
        #endregion DefineWord

        #region AddWordToHistoryBar
        private void AddWordToHistoryBar(string word)
        {
            HistoryItem item = new HistoryItem(word);
            item.Dock = DockStyle.Top;
            //item.Click += new EventHandler(HistoryItem_Click);

            //mWordHistoryLayoutPanel.Controls.Add(item);

            EnsureHistoryState();
        }
        #endregion AddWordToHistoryBar

        #region EnsureHistoryState
        private void EnsureHistoryState()
        {
            //mClearHistoryLinkLabel.Enabled = (Settings.Default.WordHistory != null &&
Settings.Default.WordHistory.Count > 0);
```

```
        }
        #endregion EnsureHistoryState

        #region RebindWordHistory
        private void RebindWordHistory()
        {
            if (Settings.Default.WordHistory != null)
            {
                //mWordHistoryLayoutPanel.Controls.Clear();
                foreach (string word in Settings.Default.WordHistory)
                {
                    HistoryItem item = new HistoryItem(word);
                    item.Dock = DockStyle.Top;
                    item.Click += new EventHandler(HistoryItem_Click);

                    //mWordHistoryLayoutPanel.Controls.Add(item);
                }
            }

            EnsureHistoryState();
        }
        #endregion RebindWordHistory

        #region ClearWordHitory
        public void ClearWordHitory()
        {
            //Settings.Default.WordHistory.Clear();
            //Settings.Default.Save();
            RebindWordHistory();
        }
        #endregion ClearWordHitory

        #region Event Handlers

        #region mGoButton_Click
        public void WordDefinition()
        {
            try
            {
                if (!string.IsNullOrEmpty(mWordTextBox.Text))
                {
                    this.Cursor = Cursors.WaitCursor;

                    DefineWord(mWordTextBox.Text);

                    this.Cursor = Cursors.Default;
                }
                else
                {
                    MessageBox.Show(this, "Please enter a word.", this.Text,
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(this, ex.Message, this.Text, MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }

        }
        private void mGoButton_Click(object sender, EventArgs e)
        {
            try
            {
                ClearWordHitory();
                if (!string.IsNullOrEmpty(mWordTextBox.Text))
                {
                    this.Cursor = Cursors.WaitCursor;

                    DefineWord(mWordTextBox.Text);
```

```
                this.Cursor = Cursors.Default;
            }
            else
            {
                MessageBox.Show(this, "Please enter a word.", this.Text,
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(this, ex.Message, this.Text, MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
    #endregion mGoButton_Click

    #region mWebBrowser_Navigating
    private void mWebBrowser_Navigating(object sender, WebBrowserNavigatingEventArgs
e)
    {
        string url = e.Url.ToString();

        if (url.StartsWith("define"))
        {
            e.Cancel = true;

            string[] segments = url.Split(':');
            if (segments.Length > 1)
            {
                string word = segments[1];
                mWordTextBox.Text = word;
                DefineWord(word);
            }
        }
    }
    #endregion mWebBrowser_Navigating

    #region HistoryItem_Click
    private void HistoryItem_Click(object sender, EventArgs e)
    {
        string word = ((HistoryItem)sender).Text;
        mWordTextBox.Text = word;
        DefineWord(word);
    }
    #endregion HistoryItem_Click

    #region mClearHistoryLinkLabel_LinkClicked
    private void mClearHistoryLinkLabel_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        //ClearWordHitory();
    }
    #endregion mClearHistoryLinkLabel_LinkClicked

    #region mAboutLinkLabel_LinkClicked
    private void mAboutLinkLabel_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
        AboutDialog about = new AboutDialog();
        about.ShowDialog(this);
    }
    #endregion mAboutLinkLabel_LinkClicked

    #endregion Event Handlers
  }
}
```

References

REFRENCES

1. Fayyad, U., Shapiro, G. P. and Smyth, P. (1996) From Data Mining to Knowledge Discovery in Databases. *American Association for Artificial Intelligence,* pp:0738-4602.
2. Kumar, V. and Chadha, A. (March 2011) An Empirical Study of the Applications of Data Mining Techniques in Higher Education, *International Journal of Advanced Computer Science and Applications (IJACSA),* Vol. 2, No.3, pp: 22-31.
3. Han, j., Cheng, H., Xin, D., and Yan, X. (2007) Frequent pattern mining: current status and future Directions, *international conference on data mining and knowledge discovery,* pp:55-86.
4. Vaidya, j. and Clifton, C. (2002) Privacy Preserving Association Rule Mining in Vertically Partitioned Data, *Second International Conference on Knowledge Discovery, Edmonton, Alberta, Canada.*
5. Han, J. and Kamber, M. (2011) *Data Mining Concepts and Techniques,* Morgan Kaufmann, USA.
6. Zamir, O. and Etzioni, O. (1998) Web document clustering: a feasibility demonstration, *Proceedings of the 21st ACM-SIGIR Conference*, pp: 46-54.
7. Tseng, Y. H., Lin, C. J., Chen, H. H. and Lin, Y. I (2006) Toward Generic Title Generation for Clustered Documents, *international conference on Alliance of Information and Referral System (AIRS),* pp: 145-157.
8. Popsecul, A., Unger, L. (2002) Automatic Labeling of Document Clusters, Retrieved from web source: http://scholar.google.com.pk/scholar.
9. Carmel, D., Roitman, H. and Zwerdling, N. (2009) Enhancing Cluster Labeling Using Wikipedia, *international conference on Special Interest Group on Information Retrieval (SIGIR), Boston, USA,* 139-146.

10. Cutting, D. R., Karger, D. R. (1992) A Cluster-based Approach to Browsing Large Document Collection, *Proceeding of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, pp: 318-329.*
11. Schneider K. M. (2009) Weighted Average Pointwise Mutual Information for Feature Selection in Text Categorization, Retrieved from *web source:* http://scholar.google.com.pk/scholar.
12. Ahmed, R., Khanum, A. (2010) Document Topic Generation in Text Mining by Using Cluster Analysis with EROCK, *International Journal of Computer Science and Security*, vol. (4).
13. Zhang, C. (2009) Clustering Description Extraction Based on Statistical Machine Learning, *Second International Symposium on Intelligent Information Technology Application*, pp: 22-26.
14. Linoff, G. S. (1996) Data Mining Techniques, Retrieved from *web source:* http://google.com.pk.
15. Ali, M. U. (2006) *Development of Benchmarking Standard for Document Clustering Algorithm based on Clustering Techniques for Data Mining*, unpublished available at: Central Library International Islamic University Islamabad (IIUI).
16. Witten, I. H. (2003) Text mining, Retrieve from *web source:* http://google.com.pk.
17. Phyu, T. N. (2009) Survey of Classification Techniques in Data Mining, *Proceedings of the International Multi Conference of Engineers and Computer Scientists, Hong Kong*: pp: 978-988.
18. Singh, Y., Chauhan, A. S. (2009) Neural Networks in Data Mining, *Journal of Theoretical and Applied Information Technology, vol. (3).*
19. Hsu, W. (2000) Genetic Algorithm available at: http://google.com.pk.
20. Treeratpituk, P., Callan, J., (2006), Automatically Labeling Hierarchical Clusters. *Second International Symposium on Intelligent Information Technology Application*, 167-176.
21. Cover, T. M., and Thomas, J. A. (1991) Elements of Information Theory, available at: http://google.com.pk.

22. Maqbool, O., Babri, H. A. (2005) Interpreting Clustering Results through Cluster Labeling, *International Conference on Engineering Technologies", Pakistan*, pp: 429-433.
23. Mei, Q., Shen, X. and Zhai, C. (2007) Automatic Labeling of Multinomial Models, 7th *international conference on Knowledge Discovery (KDD), California, USA.*
24. Guha, S., Rastogi, R. and Shim, K. (1999) ROCK, A Robust Clustering Algorithm for Categorical Attributes. *International conference on Data Engineering, Sydney, PP: 182-186.*
25. Zibran, M. F. (2000) CHI-Squared Test of Independence, available at: http://google.com.pk
26. Tous, R. and Delgado, J. (2006) A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment, *International conference on Data Engineering, Denmark, pp: 307-316.*
27. Romesburg, H. Charles, k., (2004), *Cluster Analysis for Researchers*, available at http://www.Wikipedia.com.
28. Yun-tao, Z., Ling, G. (2004) An Improved TF-IDF Approach for Text Classification, *International Journal of Zhejiang University SCIENCE, vol. (1),pp:* 39-55.
29. Sameh, A., Kadray, A., (2010) *Semantic Web Search Results Clustering Using Lingo and WordNet, International Journal of Research and Reviews in Computer Science (IJRRCS),* vol. (1), pp: 25-40.
30. Rowell, L., (2005), Conducting a Reflective Interview: Example from USD model, available at: http://google.com.pk.
31. Zhang, C. (2009) Document Cluster Description Based on Combination Strategy, 4th *international conference on innovative computing information and control, pp: 1084-1088.*
32. Roulstan, K. (2010) Reflective Interviewing: A Guide to Theory and Pratice, McGraw Hills, USA.
33. Richard F. and Ebrahim P. (2007) *Final Project Using WordNet and Clustering for Semantic Role Labeling* unpublished available at: http://google.com.pk