ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

An Automatic Labeling of the Clusters in Forensic Analysis for Improving Computer Inspection

Mrs.K.Shankari[1], Mrs.C.Rathika Prakash[2]

Research Scholar, Department of Computer Science, SRCW, Coimbatore, India[1]

Assistant Professor, Department of Computer Science, SRCW, Coimbatore, India[2]

Abstract: Computer Forensics analysis is defined as the discipline that combines elements of law and computer science which used to analysis the seized computers in Forensics department. Clustering algorithms are typically used for exploratory data analysis, where there is little or no prior knowledge about the data. This is exclusively in a number of applications of Computer Forensics, including the one addressed in our work. In exacting, algorithms for clustering documents can make possible the innovation and functional knowledge from the documents under analysis. To be had an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigations. It can be moving out with six familiar clustering algorithms (K-means,K-medoids, Single Link, Complete Link, Average Link, and CSPA) applied to five real-world datasets obtained from computers seized in real-world investigations. Automatically labeling document clusters with words which identify their topics is difficult to do well. In order to solve this problem we present two methods of labeling document clusters provoked by the model that words are generated by a hierarchy of mixture components of varying generality. The first method assumes existence of a document hierarchy (manually constructed or resulting from a hierarchical clustering algorithm) and uses a chi squared test of consequence to detect different word usage across categories in the hierarchy. The second method selects words which equally occur frequently in a cluster and effectively differentiate the given cluster from the other clusters. We compare these methods on abstracts of documents selected from a subset of the hierarchy of the Cora search engine for computer science research papers. Labels produced by our methods showed superior results to the commonly employed methods.

Keywords: Data mining, Forensic Analysis, Clustering, Fuzzy c-means, EM Algorithm

I. Introduction

Data mining is the process of extracting or mining knowledge from large amount of data. It is an analytic process designed to explore large amounts of data in search of consistent patterns and systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. It can be viewed as a result of natural evolution of information in development of functionalities such as data collection, database creation, data management, data analysis.The data mining is a step in the knowledge discovery process. The data mining step interacts with a user or a knowledge base. Computer forensics involves the preservation, identification, extraction and documentation of computer evidence stored in the form of magnetically, optically, or electronically stored media. It is a relatively new science that is becoming increasingly important as criminals aggressively expand the use of technology in their enterprise of illegal activities. Computer forensic techniques are not as advanced as those of the more mature and mainstream forensics techniques used by law enforcement, such as blood typing, ballistics, fingerprinting, and DNA testing. Its immaturity is partly attributable to fast-paced changes in computer technology, and the fact that it is a multidisciplinary subject, involving complicated associations between the legal system, law enforcement, business management, and information technology.

The volume of data in the digital world increased from 161 hexabytes in 2006 to 988 hexabytes in 2010 about 18 times the amount of information present in all the books ever written—and it continues to grow exponentially. This large amount of data has a direct impact in *Computer Forensics*, which can be broadly defined as the discipline that combines elements of law and computer science to collect and analyze data from computer systems in a way that is admissible as evidence in a court of law. In our exacting application domain, it usually involves examining hundreds of thousands of files per computer. This activity exceeds the expert's ability of analysis and interpretation of data. Therefore, methods for automated data analysis, like those widely used for machine learning and data mining, are of paramount importance. In particular, algorithms for pattern recognition from the information present in text documents are promising, as it will hopefully become evident later in the paper.

Clustering algorithms are usually used for examining data analysis, where there is slight or no prior knowledge about the data. This is accurately the case in several applications of *Computer Forensics*, including the one addressed in our work. From a more technical viewpoint, our datasets consist of unlabeled objects—the classes or categories of documents that can be found are *a priori* unknown. Moreover, even assuming that labeled datasets could be available from previous analyses, there is almost no hope

DOI 10.17148/IJARCCE.2015.4310

that the same classes (possibly learned earlier by a classifier in a supervised learning setting) would be still valid for the upcoming data, obtained from other computers and associated to different investigation processes. More precisely, it is likely that the new data sample would come from a different population. In this context, the use of clustering algorithms, which are capable of finding latent patterns from text documents found in seized computers, can enhance the analysis performed by the expert examiner.

The rationale behind clustering algorithms is that objects within a valid cluster are more similar to each other than they are to objects belonging to a different cluster. Thus, once a data partition has been induced from data, the expert examiner might initially focus on reviewing representative documents from the obtained set of clusters. Then, after this preliminary analysis, he may eventually decide to scrutinize other documents from each cluster. By doing so, one can avoid the hard task of examining all the documents (individually) but, even if so desired, it still could be done.

In a more practical and realistic scenario, domain experts (e.g., forensic examiners) are scarce and have limited time available for performing examinations. Thus, it is reasonable to assume that, after finding a relevant document, the examiner could prioritize the analysis of other documents belonging to the cluster of interest, because it is likely that these are also relevant to the investigation. Such an approach, based on document clustering, can indeed improve the analysis of seized computers, as it will be discussed in more detail later. Clustering algorithms have been studied for decades, and the literature on the subject is huge. Therefore,decided to choose a set of (six) representative algorithms in order to show the potential of the proposed approach, namely: the partitional K-means and K-medoids, the hierarchical Single/Complete/Average Link, and the cluster ensemble algorithm known as CSPA.

II. LITERATURE SURVEY

Alexander Strehl and Joydeep Ghosh, they introduce the problem of combining multiple partitioning of a set of objects without accessing the original features. Here call this the cluster ensemble problem, and will motivate this new, constrained formulation shortly. Note that since the combiner can only examine the cluster label but not the original features, this is a framework for knowledge reuse. The cluster ensemble design problem is more difficult than designing classifier ensembles since cluster labels are symbolic and so one must also solve a correspondence problem. In addition, the number and shape of clusters provided by the individual solutions may vary based on the clustering method as well as on the particular view of the data available to that method. Moreover, the desired number of clusters is often not known in advance. In fact, the `right' number of clusters in a data-set often depends on the *scale* at which the data is inspected, and sometimes equally valid (but substantially different) answers can be obtained for the same data.

Sergio Decherchi et al., they proposed that two-steps investigative process is based on (1) textual information extraction and (2) textual data analysis via clustering-based text mining tools. Textual information extraction evolves in two phases, and aims at generating a collection of raw text file from information stored in digital devices. The first step involves well-known digital forensics techniques, designed for bit stream acquisition and early analysis; the second step consists instead in textual information extraction from relevant files previously found. This work addresses text clustering for forensics analysis based on a dynamic, adaptive clustering model to arrange unstructured documents into content-based homogeneous groups. The approach is validated by using the publicly available Enron emails database as experimental domain. The research presented here shows that the document clustering framework can find consistent structures suitable for investigative issues that can considerably aid the analyst during the inquiry activity.

KilianStoffel et al., they proposed a methodology for automatically constructing, starting from forensic data, expert-system-like if-then rules. These rules should fulfil two main constraints. On one hand they should be as accurate as possible and on the other hand they should be easily understandable by a human domain expert (not necessarily a specialist in expert systems). In order to achieve these goals, decided to base our approach essentially on the methods presented in the previous section i.e. on the fuzzy inference systems and on the fuzzy clustering. The methodology, are proposing is one of the many used for inferring membership functions for fuzzy variables from raw data. The overall procedure consists of three main steps:

1) Clustering the raw data
2) Extract the membership functions from the data
3) Create the fuzzy inference system

Gerard Salton and Christopher Buckley, they proposed that the main function of a term-weighting system is the enhancement of retrieval effectiveness. Effective retrieval depends on two main factors: one, items likely to be relevant to the user’s needs must be retrieved; two, items likely to be extraneous must be rejected. Two measures are normally used to assess the ability of a system to retrieve the relevant and reject the non-relevant items of a collection, known as *recall* and *precision,* respectively. Recall is the proportion of relevant items retrieved, measured by the ratio of the number of relevant retrieved items to the total number of relevant items in the collection; precision, on the other hand, is the proportion of retrieved items that are relevant, measured by the ratio of the number of relevant retrieved items to the total number of retrieved items.

III.METHODOLOGY

Several applications of Computer Forensics, including the one addressed in our work. From a more technical viewpoint, our datasets consist of unlabeled objects the classes or categories of documents that can be found are apriori unknown. Moreover, even assuming that labeled

DOI 10.17148/IJARCCE.2015.4310

datasets could be available from previous analyses, there is almost no hope that the same classes (possibly learned earlier by a classifier in a supervised learning setting) would be still valid for the upcoming data, obtained from other computers and associated to different investigation processes. More precisely, it is likely that the new data sample would come from a different population. In this context, the use of clustering algorithms, which are capable of finding latent patterns from text documents found in seized computers, can enhance the analysis performed by the expert examiner. In our current work, employ sixteen instantiations of algorithms. In addition, provide more insightful quantitative and qualitative analyses of their experimental results in our application domain.

The following are the module that are been adopted in the methodology.

- Pre-Processing Steps
- Estimating the Number of Clusters from Data
- Clustering algorithm
- Dealing with Outliers

A. Pre-Processing Steps

Before running clustering algorithms on text datasets, performed some pre-processing steps. In particular, *stop words* (prepositions, pronouns, articles, and irrelevant document metadata) have been removed. Also, the Snowball *stemming* algorithm for Portuguese words has been used. Then, adopted a traditional statistical approach for text mining, in which documents are represented in a vector space model. In this model, each document is represented by a vector containing the frequencies of occurrences of words, which are defined as delimited alphabetic strings, whose number of characters is between 4 and 25. A dimensionality reduction technique known as Term Variance (TV) that can increase both the effectiveness and efficiency of clustering algorithms. TV selects a number of attributes (in our case 100 words) that have the greatest variances over the documents. In order to compute distances between documents, two measures have been used, namely: cosine-based distance and Levenshtein- based distance. The later has been used to calculate distances between file (document) names only.

B. Estimating the Number of Clusters from Data

In order to estimate the number of clusters, a widely used approach consists of getting a set of data partitions with different numbers of clusters and then selecting that particular partition that provides the best result according to a specific quality criterion (e.g., a relative validity index). Such a set of partitions may result directly from a hierarchical clustering dendrogram or, alternatively, from multiple runs of a partitional algorithm (e.g., K-means) starting from different numbers and initial positions of the cluster prototypes. For the moment, let us assume that a set of data partitions with different numbers of clusters is available, from which choose the best one—according to some relative validity criterion. Note that, by choosing such a data partition, are performing model selection and, as an intrinsic part of this process, are also estimating the number of clusters. A widely used relative validity index is the so-called *silhouette*, which has also been adopted as a component of the algorithms employed in our work. Therefore, it is helpful to define it even before address the clustering algorithms used in our study.

C. Clustering algorithm

The clustering algorithms adopted in our study the partitional K-means and K-medoids, the hierarchical Single/Complete/Average Link, and the cluster ensemble based algorithm known as CSPA are popular in the machine learning and data mining fields, and therefore they have been used in our study. Nevertheless, some of our choices regarding their use deserve further comments. For instance, K-medoids is similar to K-means. However, instead of computing centroids, it uses medoids, which are the representative objects of the clusters. This property makes it particularly interesting for applications in which (i) centroids cannot be computed; and (ii) distances between pairs of objects are available, as for computing dissimilarities between names of documents with the Levenshtein distance.

The CSPA algorithm essentially finds a consensus clustering from a cluster ensemble formed by a set of different data partitions. More precisely, after applying clustering algorithms to the data, a similarity (co association) matrix is computed. Each element of this matrix represents pair-wise similarities between objects. The similarity between two objects is simply the fraction of the clustering solutions in which those two objects lie in the same cluster. Later, this similarity measure is used by a clustering algorithm that can process a proximity matrix—e.g., K-medoids—to produce the final consensus clustering. The sets of data partitions (clustering's) were generated in two different ways: (a) by running K-means 100 times with different subsets of attributes (in this case CSPA processes 100 data partitions); and (b) by using only two data partitions, namely: one obtained by K-medoids from the dissimilarities between the file names, and another partition achieved with K-means from the vector space model. In this case, each partition can have different weights, which have been varied between 0 and 1 For the hierarchical algorithms (Single/Complete/Average Link), simply run them and then assess every partition from the resulting dendrogram by means of the silhouette. Then, the best partition is taken as the result of the clustering process.

For each partitional algorithm (K-means/medoids), execute it repeatedly for an increasing number of clusters. For each value of, a number of partitions achieved from different initializations are assessed in order to choose the best value of and its corresponding data partition, using the Silhouette and its simplified version, which showed good results and is more computationally efficient. In our experiments, assessed all possible values of in the interval, where is the number of objects to be clustered.

DOI 10.17148/IJARCCE.2015.4310

D. Dealing With Outliers

A simple approach to remove outliers. This approach makes recursive use of the *silhouette*. Fundamentally, if the best partition chosen by the silhouette has singletons(i.e., clusters formed by a single object only), these are removed. Then, the clustering process is repeated over and over again—until a partition without singletons is found. At the end of the process, all singletons are incorporated into the resulting data partition (for evaluation purposes) as single clusters.

E. ISSUES

- The assignment of labels to clusters done by expert examiner
- Expert label is some time more complex because of induced overlapping partitions

IV.PROPOSED MODEL

The most commonly used method, labeling with the most frequent words in the clusters, ends up using many words that are virtually void of descriptive power even after traditional stop words are removed. Another method, labeling with the most predictive words, often includes rather obscure words. Present two new ways of selecting words for cluster labeling that promise to avoid the aforementioned problems. The first method assumes the existence of a document hierarchy, either manually constructed and/or populated, or a hierarchy resulting from application of a hierarchical clustering algorithm. Using chi squared tests of independence at each node in the hierarchy starting from the root determine a set of words that are equally likely to occur in any of the children of a current node. Such words are general for all of the sub trees of a current node, and are excluded from the nodes below. The second method selects words which both occur frequently in a cluster and effectively discriminate the given cluster from the other clusters.

A. Chi squared Method

The Chi squared test is well suited for testing dependencies when count data is available. The main idea of our method is to use Chi squared tests for each word at each node in a hierarchy starting at the root and recursively moving down the hierarchy. If one cannot reject the hypothesis that a word is equally likely to occur in all of the children of a given node, it is marked as general to the current sub tree, assigned to the current node's bag of node− specific words and removed from all nodes under the current node.

$$\chi^2 = \sum \frac{(\text{Observed Value - Expected Value})^2}{(\text{Expected Value})}$$

The detailed description of the algorithm follows:

Input: A hierarchy of documents where the leaves contain bags of words from all of the documents in that leaf unioned together

1. Populate all internal nodes by unioning the bags of words in its children starting from the leaves and moving up to the root;

2. Start at the root and for each word perform a Chi squared test to discover dependencies:

- If a test rejects the independence hypothesis, conclude that the word has different probability of occurring in children and thus is specific to one or more categories down the tree;
- If a test fails to reject the independence hypothesis, conclude that the word is equally likely to occur in all of the children. Retain the word at the current node as being general to the sub tree rooted at the current node. Remove all such words from all of the nodes below the node at which the test was performed.

3. Repeat step 2 recursively moving down the tree to the leaves.

Output: A hierarchy of words isomorphic to the initial hierarchy of documents, where each node contains words specific to that node and not occurring in the sub tree below the current node.

A label is a list of the most frequent words at the node corresponding to a cluster of documents wants to label.

B. Frequent and Predictive Words Method

The "frequent and predictive words" method, words are selected as labeling based on the product of local frequency and predictiveness:

$$p(\text{word} \mid \text{class}) \times \frac{p(word \mid class)}{p(word)}$$

This combined use of local frequency and predictiveness was used by Yarowsky to select the most important words in categories for illustrating his approach of word sense disambiguation. As far as , this method has not been used to label document clusters.The formula consists of two parts each having a well defined meaning: the first term, predictiveness, **p(word | class) / p(word)** is similar to a mutual information estimator and TF−IDF measure used in information retrieval in that is distributes more weight to the words occurring frequently in a given cluster and less weight to the words occurring frequently in all of the clusters; **p(word | class)** is frequency of the word in a given cluster and **p(word)** is the word's frequency in a more general category or in the whole collection. Words receiving high predictiveness values are good discriminators in distinguishing one cluster from another. Words selected by this formula tend to both occur often in a cluster and be specific to the cluster. This avoids the dilution of a label by generally frequent words and by words that are obscure. One might think of selecting the most predictive words, subject to the constraint that they be statistically significant, or appear a minimum number of times, or be also on the list of frequent words. This gives less good results than taking the product of predictiveness and frequency, which does better at selecting words high on both scales.

C. Fuzzy C-Means (FCM) clustering

Fuzzy c-means (FCM) is a method of clustering which allows one student data to belong to two or more clusters. It is based on following objective function:

DOI 10.17148/IJARCCE.2015.4310

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m}\left\|x_i - c_j\right\|^2 \quad , \quad 1 \le m < \infty$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of measured data, C_j is the numeral of the students in the jthcluster and ||*|| is any norm expressing the similarity between features (scores) of the students and the center. Fuzzy separating is carried out concluded an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|}\right)^{\frac{2}{m-1}}} \quad , \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

This iteration will stop when $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$, where $\varepsilon$is a termination criterion between 0 and 1, whereas k are the repetition steps. This technique converges to a local smallest or a saddle point of J_m.

Objective: Obtain the clusters of the given data sets.

Input:

X: set of data points

C: set of centres

m: any value from 1 to infinity

c: number of cluster centres

u: fuzzy membership

Output: Clustered data set

The algorithm is collected of the subsequent steps:

1. First, initialize U=[Uij]matrix , $U^{(0)}$
2. Calculate the midpoint vectors at k step $C^{[K]}$ =[C_j] with $U^{(k).,}$
3. Update the membership $U^{(k)}$, $U^{(k+1)}$
4. u_{ij} is the degree of membership of x_i in the cluster.
5. The update of membership u_{ij} and the cluster centers c_j by

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|}\right)^{\frac{2}{m-1}}} \quad , \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

6. The iteration will stop when $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$, where $\varepsilon$is a termination criterion between 0 and 1, whereas k are the repetition steps.

In the FCM approach, instead, the same given datum does not belong exclusively to a well-defined cluster, but it can be placed in a middle way. In this case, the membership function follows a flatter line to designate that every datum might go to numerous clusters with dissimilar standards of the membership constant.

D .EM Algorithm

The EM algorithm is used to find the maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Finding a maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values viz. the parameters and the latent variables and simultaneously solving the resulting equations. In statistical models with latent variables, this usually is not possible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice-versa, but substituting one set of equations into the other produces an unsolvable equation.

The EM algorithm proceeds from the observation that the following is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work at all, but in fact it can be proven that in this particular context it does, and that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a maximum or a saddle point. In general there may be multiple maxima, and there is no guarantee that the global maximum will be found. Some likelihood also have singularities in them, i.e. nonsensical maxima. For example, one of the "solutions" that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points.

Given a statistical model consisting of a set $\mathbf{X}$ of observed data, a set of unobserved latent data or missing value $\mathbf{Z}$, and a vector of unknown parameters $\boldsymbol{\theta}$, along with a likelihood function $L(\boldsymbol{\theta};\mathbf{X},\mathbf{Z}) = p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$, the maximum likelihood estimation (MLE) of the unknown parameters is determined by the marginal likelihood of the observed data

$$L(\boldsymbol{\theta};\mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$$

However, this quantity is often intractable (e.g. if $\mathbf{Z}$ is a sequence of events, so that the number of values grows exponentially with the sequence length, making the exact calculation of the sum extremely difficult).

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying the following two steps:

Step 1: Expectation step (E step): Calculate the expected value of the log likelihood function, with respect to the condition distribution of $\mathbf{Z}$ given $\mathbf{X}$ under the current estimate of the parameters $\boldsymbol{\theta}^{(t)}$:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \operatorname{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)}}\left[\log L(\boldsymbol{\theta};\mathbf{X},\mathbf{Z})\right]$$

Step 2: Maximization step (M step): Find the parameter that maximizes this quantity:

$$\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\arg\max}\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

The motivation is as follows. If , the value of the parameters $\boldsymbol{\theta}$, can usually find the value of the latent variables $\mathbf{Z}$ by maximizing the log-likelihood over all possible values of $\mathbf{Z}$, either simply by iterating over $\mathbf{Z}$ or through an algorithm such as the Viterbi algorithm for hidden markov model Conversely, if the value of the latent variables $\mathbf{Z}$, an estimate of the parameters $\boldsymbol{\theta}$ fairly easily, typically by simply grouping the observed data points according to the value of the associated latent variable and averaging the values, or some function of the values, of the points in each group.

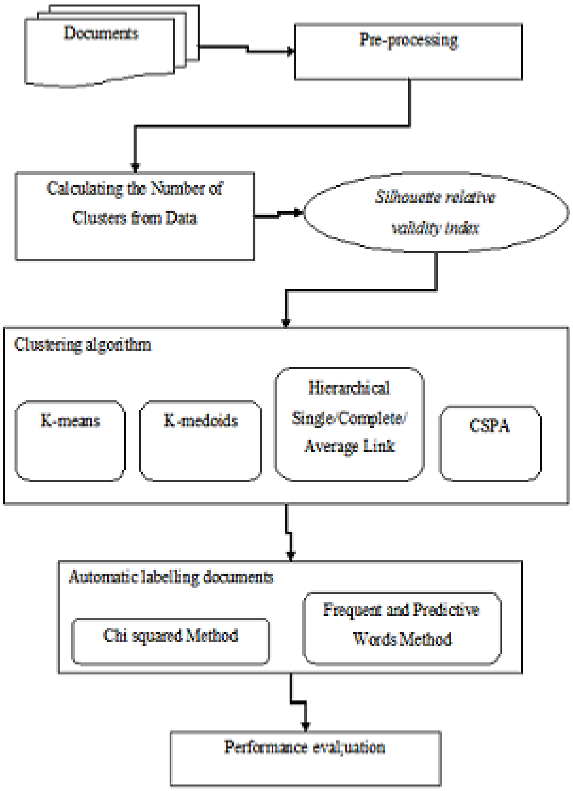
Documents
Pre-processing
Calculating the Number of Clusters from Data
Silhouette relative validity index
Clustering algorithm
K-means
K-medoids
Hierarchical Single/Complete/ Average Link
CSPA
Automatic labelling documents
Chi squared Method
Frequent and Predictive Words Method
Performance eval;uation

Figure 1:System Architecture

The proposed methods used for an automatic labeling of the clusters in forensic analysis. This proposed work is done by using MATLAB software.

The flow of proposed system architecture are described and the algorithm used in the proposed work.

V. EXPERIMENTAL SETUP

In this section, analyze and compare the performance offered by K-means, K-medoids, Single Link, Complete Link, Average Link, and CSPA, FCM, EM, automatic labeling technique.

The performance is evaluated by the parameters such as accuracy, precision, recall and f-measure. Based on the comparison and the results from the experiment show the proposed approach.

A. Data Sets

Table 1:Dataset

DATASET	CATEGORIES
20NG-Binary	talk.politics.mideast, talk.politics.misc
20NG-Multi5	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast
20NG-Multi10	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns
20NG-Diff4	comp.graphics, rec.sport.baseball, sci.space, talk.politics.mideast
20NG-Sim4	comp.graphics, comp.os.ms-windows.misc, rec.autos, sci.electronics
20NG-Long	comp./, sci./, talk./

\

The proposed representation model and classification framework were tested on real data, 20Newsgroups. Six subsets were extracted from 20Newsgroups: 20NGDiff4, 20NG-Sim4, 20NG-Binary, 20NG-Multi5, 20NG-Multi10 and 20NG-Long.

Table 2: Dataset Summary

DATASET	CLASSES	DOCUMENTS	WORDS	CONCEPTS
20NG-Binary	2	500	3376	2987
20NG-Multi5	5	500	3310	2735
20NG-Multi10	10	500	3344	2772
20NG-Diff4	4	4000	5433	4362
20NG-Sim4	4	4000	4352	3502
20NG-Long	3	210	4244	3738

Tables 1 and 2 list the categories and the number of documents contained in these subsets. In this paper, 20NG-Long is a collection of long documents containing three categories ‘‘comp’’, ‘‘sci’’ and ‘‘talk’’. In each category, 70 documents with the most large size were extracted from the corresponding topic in 20Newsgroups (documents from topic ‘‘rec’’ were not included because there are few long documents in ‘‘rec/’’). In 20NG-long, the minimal document’s size is 10 K, the maximal one is 158 KB and the average size is 29 KB.

VI. RESULTS AND DISCUSSION

This section presents the experiment results analyze and compare the performance offered by K-means, K-medoids, Single Link, Complete Link, Average Link, and CSPA, FCM, EM, automatic labelling technique. The performance is evaluated by the parameters such as accuracy, precision, recall and f-measure. Based on the comparison and the results from the experiment show the proposed approach works better than the presented method.

DOI 10.17148/IJARCCE.2015.4310

A. Accuracy

Accuracy can be calculated from formula given as follows

$$\textbf{Accuracy} = \frac{\textbf{True positive + True negative}}{\textbf{True positive + True negative+False positive+ False negative}}$$

The accuracy rate of the presented K-means, K-medoids, Single Link, Complete Link, Average Link, CSPA and proposed FCM, EM, automatic labelling technique based on two parameters of accuracy and methods such as presented and proposed system. From the graph, accuracy of the system is reduced somewhat in presented method than the proposed system. From this graph the accuracy of proposed system is increased which will be the best one.

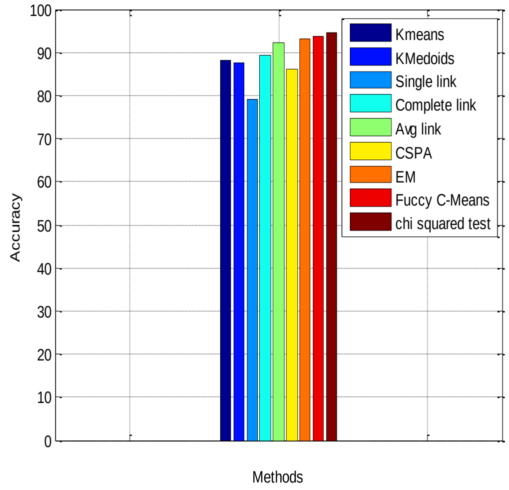

Figure 2: Accuracy Comparison

B. Precision

Precision value is calculated is based on the retrieval of information at true positive prediction, false positive .In healthcare data precision is calculated the percentage of positive results returned that are relevant.

$$\textbf{Precision} = \frac{\textbf{True positive}}{\textbf{True positive+False positive}}$$

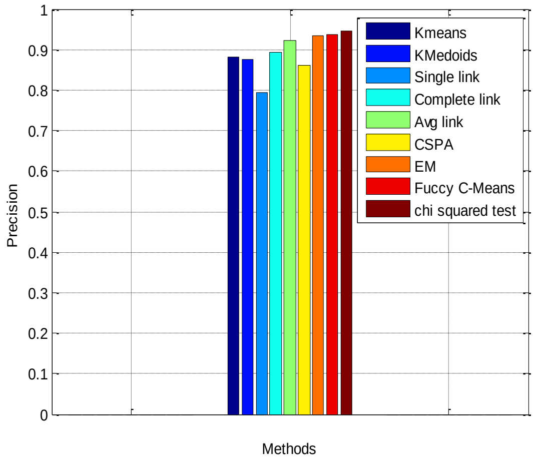

Figure 3: Precision comparison

C. Recall

Recall value is calculated is based on the retrieval of information at true positive prediction, false negative. In healthcare data precision is calculated the percentage of positive results returned that are Recall in this context is also referred to as the True Positive Rate. Recall is the fraction of relevant instances that are retrieved,

$$\textbf{Recall} = \frac{\textbf{True positive}}{\textbf{True positive+False negative}}$$

In this section, compare the recall parameter between presented K-means, K-medoids, Single Link, Complete Link, Average Link, CSPA and proposed FCM, EM, automatic labelling technique. Recall means information retrieval. It is mathematically calculated by using formula. As usual in the graph X-axis will be methods such as presented and proposed system and Y-axis will be recall rate. From view of this recall comparison graph ,obtain conclude as the proposed algorithm has more effective in recall performance compare to presented algorithms.

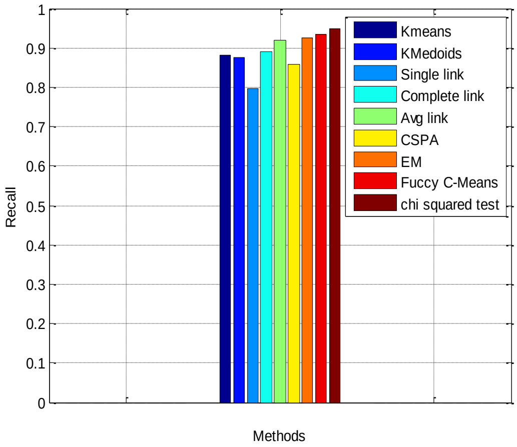

Figure 4: Recall comparison

D. F-measure Comparison

F-measure distinguishes the correct classification of document labels within different classes. In essence, it assesses the effectiveness of the algorithm on a single class, and the higher it is, the better is the clustering. It is defined as follows:

F=2.precision.recall/precision+recall

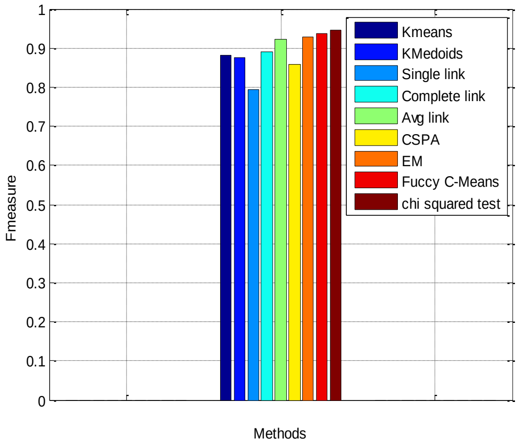

Figure 5: F-measure comparison

In this section, compare the F-measure parameter between presented K-means, K-medoids, Single Link, Complete Link, Average Link, CSPA and proposed FCM, EM, automatic labeling technique. It is mathematically calculated by using formula. As usual in the graph X-axis will be methods such as presented and proposed system and Y-axis will be F-measure rate. From view of this F-measure comparison graph, obtain conclude as the proposed algorithm has more effective in F-measure performance compare to presented method.

Table 3: Comparative Table

Metric	Accuracy	Precision	Recall	F-Measure
K means	88.2022	0.8826	0.8866	0.8841
K medoids	87.6923	0.8805	0.8762	0.8783
Single link	79.2308	0.7928	0.7926	0.7927
Complete link	89.2308	0.8925	0.8925	0.8925
Avg link	92.3077	0.9250	0.9226	0.9238
CSPA	86.1538	0.8627	0.8620	0.8623
EM	93.0769	0.9308	0.9309	0.9308
Fuccy C-means	93.8462	0.9384	0.9384	0.9384
Chi Squared Test	94.6154	0.9462	0.9463	0.9462

The results obtained for compare the F-measure parameter between the presented K-means, K-medoids, Single Link, Complete Link, Average Link, CSPA and proposed FCM, EM, automatic labeling technique. It is mathematically calculated by using formula. From view of this F-measure comparison graph, obtain conclude as the proposed algorithm has more effective in F-measure performance compare to presented method.

VII. Conclusion

This paper has concentrated on automatically labeling document clusters with words which indicate their topics are difficult to do well. In order to solve this problem, present two methods of labeling document clusters motivated by the model that words are generated by a hierarchy of mixture components of varying generality. This proposed work has presented two methods of labeling document clusters by selecting topic revealing keywords. The most frequent and predictive words method produced the best labels, capturing the words which both occur frequently in a cluster and effectively discriminate the given cluster from the other clusters, but the Chi squared method also outperformed labeling by either most frequent or most predictive words. The Chi squared method also successfully identified a set of *collection specific stop words*, words that are common to a given collection of documents, but are not part of the traditional stop word list, and lack any descriptive power to someone browsing the document collection.

The Chi squared method checks to see if word frequencies differ in any of the child nodes. This lends to poor performance in hierarchies with high branching factor. The method could be improved by checking for subsets of the child nodes where words have similar frequencies and excluding such words from these children while retaining them in the other children. Unfortunately, none of the methods gave uniformly satisfactory results at the internal nodes of the hierarchy. This could well be a feature of the document collection, showing that the disciplines corresponding to the internal nodes, Information Retrieval, Artificial Intelligence, and Machine Learning, are very diverse in the vocabulary used and encompass very broad topics.

References

[1] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz, “The expanding digital universe: A forecast of worldwide information growth through 2010,” *Inf. Data*, vol. 1, pp. 1–21, 2007.

[2] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. London, U.K.: Arnold, 2001.

[3] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[4] L. Kaufman and P. Rousseeuw, *Finding Groups in Gata: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley-Interscience, 1990.

[5] R. Xu and D. C.Wunsch, II, *Clustering*. Hoboken, NJ: Wiley/IEEE Press, 2009.

[6] A. Strehl and J. Ghosh, “Cluster ensembles: A knowledge reuse framework for combining multiple partitions,” *J. Mach. Learning Res.*, vol. 3, pp. 583–617, 2002.

[7] E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro, “Evolving clusters in gene-expression data,” *Inf. Sci.*, vol. 176, pp. 1898–1927, 2006.

[8] B. K. L. Fei, J. H. P. Eloff, H. S. Venter, andM. S. Oliver, “Exploring forensic data with self-organizing maps,” in *Proc. IFIP Int. Conf. Digital Forensics*, 2005, pp. 113–123.

[9] N. L. Beebe and J. G. Clark, “Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results,” *Digital Investigation, Elsevier*, vol. 4, no. 1, pp. 49–54, 2007.

[10] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, “Towards an integrated e-mail forensic analysis framework,” *Digital Investigation, Elsevier*, vol. 5, no. 3–4, pp. 124–137, 2009.

[11] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, “Mining writeprints from anonymous e-mails for forensic investigation,” *Digital Investigation, Elsevier*, vol. 7, no. 1–2, pp. 56–64, 2010.

[12] S. Decherchi, S. Tacconi, J. Redi, A. Leoncini, F. Sangiacomo, and R. Zunino, “Text clustering for digital forensics analysis,” *Computat. Intell. Security Inf. Syst.*, vol. 63, pp. 29–36, 2009.

[13] K. Stoffel, P. Cotofrei, and D. Han, “Fuzzy methods for forensic data analysis,” in *Proc. IEEE Int. Conf. Soft Computing and Pattern Recognition*, 2010, pp. 23–28.

[14] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, “Relative clustering validity criteria: A comparative overview,” *Statist. Anal. Data Mining*, vol. 3, pp. 209–235, 2010.

[15] [15] G. Salton and C. Buckley, “Term weighting approaches in automatic text retrieval,” *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.