ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 3 (July - Sept. 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

An Efficient Clustering Algorithm for Text Mining Using Greedy Approach

[I]T. Periyasamy, [II]M. Senthil Kumar
[I]Assistant Professor, Department of MCA, Saraswathi Thyagaraja College, Pollachi.
[II]Research Scholar, Saraswathi Thyagaraja College, Pollachi.

Abstract
Text clustering is a text mining technique used to group text documents into groups (or clusters) based on similarity of content. This organization (i.e. clustering) is so as to make documents more understandable and easier to search the relevant information, easier to process, and even more efficient in utilizing communication bandwidth and storage space. Clustering problems can be defined as: given a dataset of N records, each having dimensionality d, to partition the data into subsets such that a specific criterion is optimized. The most widely used criterion for optimization is the distortion criterion. Each record is assigned to a single cluster and distortion is the average squared Euclidean distance between a record and the corresponding cluster center. Thus this criterion minimizes the sum of the squared distances of each record from its corresponding center. A new approach has been proposed for avoiding clustering problem, which is called greedy approach. Global K-means clustering is used to minimize the above-mentioned term by partitioning the data into k non-overlapping regions identified by their centers.
K Means is arguably the most popular text clustering algorithm. However, just like the others, it must be having its own weaknesses. We explore the K Means algorithm as well as its variants and discuss their appropriateness in text clustering. The final the proposed result explains of text mining concerning the choice of Global K Means for text clustering.

Keywords
Text, Mining, Clustering, K-Means, Greedy, Global K-Means.

I. Introduction

"Data Mining" involves the integration of concepts from computer science, mathematics, and statistics. It seeks to extract useful information and detect interesting correlation and patterns from any form of data, especially numeric data. Data Mining is most associated with the broader process of Knowledge Discovery in Databases (KDD), "the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data" (Fayyad et al., 1996). By analogy, "text mining" as the process that exploits large text collections to obtain valid, potentially useful and ultimately understandable knowledge.
It is important to emphasize that getting from a collection of documents to a clustering of the collection, is not merely a single operation, but is more a process in multiple stages. These stages include more traditional information retrieval operations such as crawling, indexing, weighting, filtering etc. Some of these other processes are central to the quality and performance of most clustering algorithms, and it is thus necessary to consider these stages together with a given clustering algorithm to harness its true potential. We will give a brief overview of the clustering process, before we begin our literature study and analysis. We have divided the offline clustering process into the four stages outlined below:

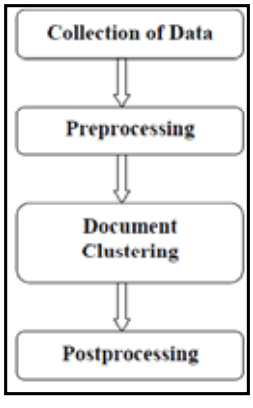

Fig 2.1: - The Stages of the process of the Clustering

Collection of Data includes the processes like crawling, indexing, filtering etc. which are used to collect the documents that needs to be clustered, index them to store and retrieve in a better way, and filter them to remove the extra data , for example, stop words. Preprocessing is done to represent the data in a form that can be used for clustering. There are many ways of representing the documents like, Vector-Model, graphical model, etc. Many measures are also used for weighing the documents and their similarities.
Document Clustering is the main focus of this thesis and will be discussed in detail. Post processing includes the major applications in which the document clustering is used, for example, the recommendation application which uses the results of clustering for recommending news articles to the users.
Text mining is a burgeoning new field that attempts to glean meaningful information from natural language text. It may be loosely characterized as the process of analyzing text to extract information that is useful for particular purposes. Compared with the kind of data stored in databases, text is unstructured, amorphous, and difficult to deal with algorithmically. Nevertheless, in modern culture, text is the most common vehicle for the formal exchange of information. The field of text mining usually deals with texts whose function is the communication of factual information or opinions, and the motivation for trying to extract information from such text automatically is compelling—even if success is only partial.

II. Text Mining Process

There are approximately five major technique categories in the text mining process: document retrieval, data extraction, data preprocessing (cleansing), data analysis, and data visualization.

A. Document Retrieval

Information (or document) retrieval is a discipline concerned with the organizing, storage, searching, and retrieval of bibliographic information. Salton and McGill (1983) introduce the idea of the

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Vector Space Model (VSM) -- a vector, comprised of the keywords contained within the document, can represent a document. It is a powerful framework for analyzing and structuring documents. VSM model procedures can be divided into three stages: document indexing, term weighting, and computation of similarity coefficients.

B. Data Extraction

Data extraction is the activity of automatically pulling out pertinent information from large volumes of texts. Extraction can take two forms; one is to identify the specific field of entity extracted such as name, date, or address, and the other one is to identify the parts of speech from text corpus using natural language processing (NLP) technology. VantagePoint applies NLP to parse text into the part(s) of speech. It employs a combination of semantic and syntactic analyses. It processes text inputs as follows.

- Distinguishes and separates each sentence.
- Applies lexicon analysis to categorize nouns, verbs, etc., based on the underlying dictionary.
- Refines word attribution based on syntactic inferences.

This then tags each word with the part(s) of speech it is likely to be.

C. Data Preprocessing

Data preprocessing, or data cleansing, is the algorithm that detects and removes errors or inconsistencies from data and consolidates similar data in order to improve the quality of subsequent analyses. This cleaned data will then be fed to the analysis process.

Several methods could be used to clean the data. Three methods that are used in VantagePoint's list cleanup are stemming algorithm, elemental fuzzy logic to consolidate like terms, and thesauri. Word stemming or truncation can be used to achieve a quick approximation to the word root. A word for which one wants to find an exact or near match may by written as a stem or root word, and the retrieval system asked to find words that match the root. One approach used to determine the root of a word is to determine the semantic root such that "box" and "boxes" are equal. Fuzzy matching techniques can be used to identify, associate, and reduce data appropriately. For example, this will handle misspellings, alternative hyphenation and capitalization. A thesaurus is defined as a grouping of terms, into a certain concepts. This can be used for specialized data reduction.

D. Data Analysis

As stated before, each document can be represented as a vector in a high dimensional space. Hence, dimensionality reduction techniques are required to represent *n*-dimensional document data by a small number of significant dimensions. There are several techniques that have been used for dimensionality reduction, including Factor Analysis (FA)/Principal Component Analysis (PCA) and Cluster Analysis.

E. Data Visualization

A general goal of analysis is to detect meaningful underlying dimensions that allow the researcher to explain observed similarities or dissimilarities (distances) among the investigated objects. This is accomplished by solving a minimization problem such that the distances among points in the conceptual low-dimensional space match the given (dis)similarities as closely as possible. In factor analysis, the similarities among objects (e.g., terms) are expressed in the correlation matrix. With MDS, one may analyze any kind of similarity or dissimilarity matrix, in addition to correlation matrices. However, a major weakness of MDS is that there are no quick and fast rules to interpret the nature of the resulting dimensions.

III. Problem Description

One main reason for applying data mining methods to text document collections is to structure them. A structure can significantly simplify the access to a document collection for a user. Well known access structures are library catalogues or book indexes. However, the problem of manual designed indexes is the time required to maintain them. Therefore, they are very often not up-to-date and thus not usable for recent publications or frequently changing information sources like the World Wide Web. The existing methods for structuring collections either try to assign keywords to documents based on a given keyword set (classification or categorization methods) or automatically structure document collections to find groups of similar documents (clustering methods). The problem of Text Mining is therefore Classification of data set and Discovery of Associations among data.

Finding relevant information in unstructured data is a challenge. The data is unknown in terms of structure and values. The lifecycle of each part of data is in a specific domain, whereby a domain expert is available for a priori knowledge. Domain experts can creates structures by hand in the data, however this is a time-consuming job and it is done for one dataset in one domain.

There are many different clustering algorithms. Most of them need to know the (dis)similarity between the objects (texts). Some of them need a representation for each object, and a definition of similarity, so they can calculate it when necessary. How the objects are represented and the definition of similarity differs between applications. It is usually convenient to build a representation and define similarity in terms of it. There are many ways to achieve this. If, for instance, the objects can be represented as points in a n-dimensional vector space, dissimilarity could be defined as the distance between them. When given a set of objects and the similarity between them, a clustering algorithm outputs a partition that tries to satisfy some criteria. It could be that the objects in each cluster should be as similar as possible. However, in order for the result to be useful at all, we must have reasons to believe that the similarity definition reflects the similarity between the actual objects.

This work focuses on content groups, that are groups of texts that are similar in content. So how do we represent texts and define (content) similarity between them? In many Information Retrieval applications texts are represented by the words that appear in them and similarity between two texts is defined by considering the words that appear in both of them. Basically, two texts that share many words are considered more similar than two that share fewer words.

IV. Existing Methodology

Clustering is a well-known problem in statistics and engineering, namely, how to arrange a set of vectors (measurements) into a number of groups (clusters). Clustering is an important area of application for a variety of fields including data mining, statistical data analysis and vector quantization.

"Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources." Text mining techniques are

the fundamental and enabling tools for efficient organization, navigation, retrieval and summarization. With more and more text information are spreading around on Internet, text mining is increasing in importance. Text clustering and text classification are two fundamental tasks in text mining.

The k-means algorithm is based on the simple observation that the optimal placement of a center is at the centroid of the associated cluster.

The attractiveness of the k-means lies in its simplicity and flexibility. In spite of other algorithms being available, k-means continues to be an attractive method because of its convergence properties. However, it suffers from major shortcomings that have been a cause for it not being implemented on large datasets. The most important among these are

1. K-means is slow and scales poorly with respect to the time it takes for large number of points;
2. The algorithm might converge to a solution that is a local minimum of the objective function.

Much of the related work does not attempt to confront both the before mentioned issues directly. Because of these shortcomings, K-means is at times used as a hill climbing method rather than a complete clustering algorithm, where the centroids are initialized to the centers obtained from some other methods.

Before proceeding, we explicitly define the problem that we are looking at. In a traditional k-means algorithm (for which we develop an improved solution) each of the points is associated with only one partition also called the cluster. The number of partitions is pre-specified. Each of the partitions is recognized by a cluster center, which is the mean of all the points in the partition.

All the points in a partition are closer to its center than to any other cluster center. The accuracy of a clustering approach is defined by the distortion criterion, which is the mean squared distance of a point from its cluster center. Thus the objective is to get the clustering with minimum distortion and the specified number of clusters.

V. Proposed Scheme

In this proposed research, the global k-means clustering algorithm is proposed, which constitutes a deterministic global optimization method that does not depend on any initial parameter values and employs the k-means algorithm as a local search procedure. Instead of randomly selecting initial values for all cluster centers as is the case with most global clustering algorithms, the proposed technique proceeds in an incremental way attempting to optimally add one new cluster center at each stage.

More specifically, to solve a clustering problem with M clusters the method proceeds as follows. We start with one cluster (k = 1) and 2nd its optimal position which corresponds to the centroid of the data set X. In order to solve the problem with two clusters (k =2) we perform N executions of the k-means algorithm from the following initial positions of the cluster centers: the first cluster center is always placed at the optimal position for the problem with k = 1, while the second center at execution n is placed at the position of the data point xn (n=1; : : : ; N). The best solution obtained after the N executions of the k-means algorithm is considered as the solution for the clustering problem with k =2. In general, let (m*1 (k); : : : ; m*k (k)) denote the final solution for k-clustering problem. Once we have found the solution for the (k − 1)-clustering problem, we try to find the solution of the k-clustering problem as follows: we perform N runs of the k-means algorithm with k clusters where each run n starts from the initial state (m*1 (k−1); : : : ; m*(k−1)(k−1); xn). The best solution obtained from the N runs is considered as the solution (m*1 (k); : : : ; m*k (k)) of the k-clustering problem. By proceeding in the above fashion we finally obtain a solution with M clusters having also found solutions for all k-clustering problems with k ¡M.

This effective method is not only deterministic but it also does not depend upon any initial conditions or empirically adjustable parameters. Above mentioned points are the significant advantages of overall clustering approaches.

The global k-means algorithm successively computes the clusters. For first iteration, the centroid of set A is computed. Similarly for computing k-partition, k-th iteration of this algorithm uses k-1 clusters centers from the previous iteration. We can describe the global k-means algorithm for the computation of q ≤ m clusters in a data set A are as follows.

Algorithm: The global k-means algorithm

Step 1: - (Initialization) Compute the centroid x1 of the set A: And set k = 1.

Step 2: - Set k = k + 1 and consider the centers x1, x2, . . . , xk−1 from the previous iteration.

Step 3: - Each point of A is the starting point for the k-th cluster center, To obtain m initial solutions with k points (x1, . . . , xk−1, a); k-means algorithm is applied to each of them; keep the best k-partition obtained and its centers x1, x2, . . . , xk.

Step 4: - (Stopping criterion) If k = q then stops, otherwise go to Step 2.

Global k-means is the incremental algorithm that allows us to add one cluster center at a time and uses each data point as a candidate for the *k*-th cluster center. Experimental results show that the global *k*-means algorithm considerably outperforms the *k*-means algorithms. New version of this algorithm is proposed in this paper, it uses minimizing an auxiliary cluster function to compute the starting point for the *k*-th cluster center. Numerical results of these experiments (i.e. 14 data sets) demonstrate the superiority of the new algorithm, however it required more computational time than global k-mean algorithm.

A. Greedy Approach

The time complexity of the algorithm can also be improved by taking a *greedy approach*. In this approach, running k-means for each possible insertion position is avoided. Instead reduction in the distortion when the new cluster is added is taken into account without actually running k-means. The point that gives the maximum decrease in the distortion when added as a cluster center is taken to be the new insertion position. K-means is run until convergence on the new list of clusters with this added point as the new cluster. The assumption is that the point that gives the maximum decrease in distortion is also the point for which the converged clusters would have the least distortion. This results in a substantial improvement in the running time of the algorithm, as it is unnecessary to run k-means for all the possible insertion positions. However, the solution may not be globally optimal but an approximate global solution.

STEP 1: Calculate the value of *maximum safety bound* (*MSB*) for the number of newly inserted transactions as:

$$n = \max_{i=1}^{p}(SB_i) = \left[\frac{|si_i|}{\alpha} - m\right] + 1$$

Where *SBi* is the *safety bound* of each sensitive itemset, *m* is the

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

number of original transactions in *D*, |*sii*| is the count of sensitive itemset *sii*.

STEP 2: Calculate the length *pn* of each inserted transaction in *d* according to the empirical rules in standard normal distribution, where *d* = *{d1; dd; : : : ; dn}*, and *n* is the number of inserted transactions obtained in STEP 1.

STEP 3: Choose the itemsets to be inserted into each inserted transaction *dn*.

STEP 4: Process the inserted transactions *dn* one-by-one respectively to add the *fik* in the set of *Insert Items* according to the sorted order obtained in substep 3-4.

STEP5: Update (decrease) the value |*CDfik* and the corresponding sub-itemsets of the processed itemset *fik* by 1.

STEP 6: Repeat the STEPs 4 to 5 until the set of *Insert Items* is *null* or there is no longer itemsets to be inserted into *dn* obtained the constraints in STEP 4.

STEP 7: Add the small items in the set of *I* into the *dn* while *dn* remains positions to be added according to empirical rules in standard normal distribution.

VI. Experimental Results

The global k-means (GKM) clustering algorithm is one of the most effective approaches for resolving the local convergence of the k-means clustering algorithm. Numerical experiments show that it can effectively determine a global or near global minimize of the cost function. The great advantage of GKM is that it is improving the way of creating the next cluster center and it defined a novel function to select the optimal candidate center for the next cluster. But GKM also have the problem that it takes much time to compute large data sets. It also takes large space to implement large data sets. When we will apply GKM of high definition (HD) data sets or streamed data sets than this will cause the same problem of storage and time.

Four data sets are used, all of which contains only numeric attributes and class attributes. The information about the data sets is tabulated in Table 6.1.

Table 6.1: - Data Set for Experiment

Data Set	Size	Attribute	Class
Cancer	28	10	2
Pen Digits	724	17	10
Spice	511	62	3
Vehicle	64	19	4

We have proposed a modification in the simple K-means algorithm and the experiments prove that with this modification the clustering performance drastically increase, by changing the distance similarity measure. The performance of proposed algorithm is tested across four real world datasets and the results are quite encouraging and have established the effectiveness of the proposed algorithms.

i) Convergence Rate for GKM

This algorithm was coded so that the stopping parameter was defined as the number of iterations to run the algorithm, regardless of whether the algorithm converged or not. This allowed for an interesting experiment of how many iterations are required to obtain convergence, given different dimensional and data sizes.

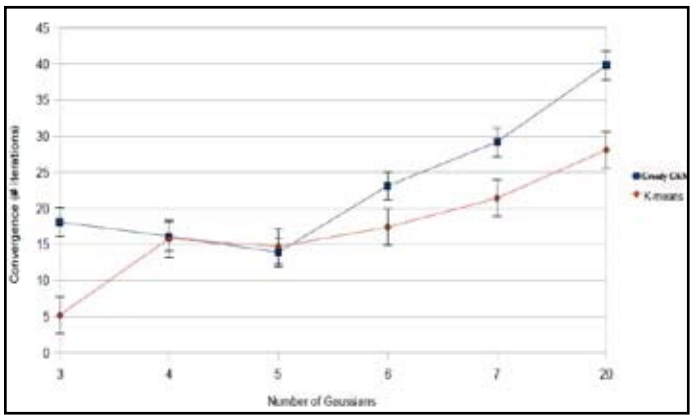

Fig 6.1: - Convergence rate of RKM versus Standard K-means.

ii) Error Rate

The data sets are taken from random Gaussian distributions (500 datum per Gaussian). GKM ran until convergence with λ=0.5 and s=1. "Standard" is the standard K-means algorithm with random initial centroid placement, "Distance" is K-means using a heuristic that places initial centroids far away from each other, "Point" places initial centroids directly on random points, and "Repeat" repeats the standard K-means algorithm 3 times and takes the best result.

Fig 6.2: - Average error rates of GKM & K-Means

iii) Execution Time

To test the efficiency in execution time of GKM, relative to the other clustering algorithms, repeatedly executed each algorithm on data sets generated from random Gaussian distributions. Averaged out the execution time and tested it for different size data sets. Each Gaussian contains 500 data points, so the size of each data set also increases with the number of Gaussians/centroids.

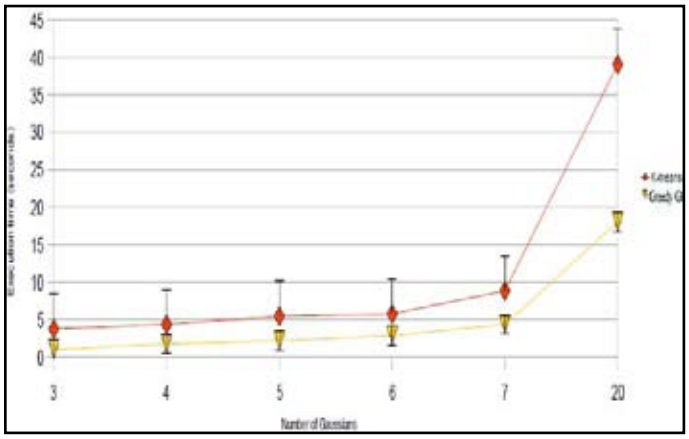

Fig 6.3 : Execution time of K-means versus GKM

iv) Number of Clusters

We consider a wide range of problems in our experiment. The generated were classes to cluster evaluation and found the higher accuracy of these clusters were observed. Therefore, this research contributed for Global K-means algorithm as a faster algorithm and optimal clustering performance.

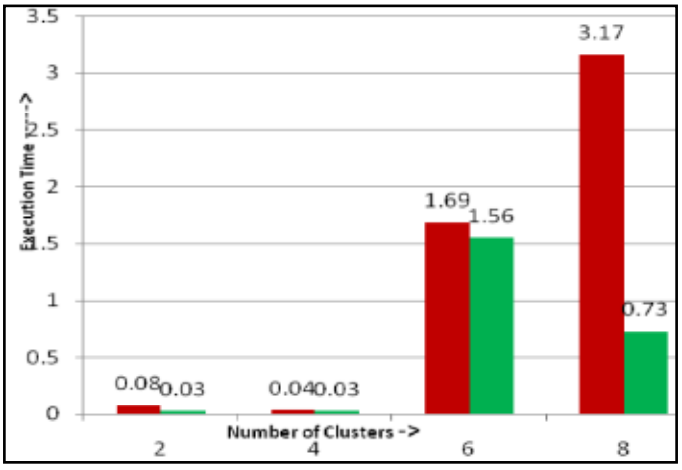

Fig. 6.4: Average No. of Clusters & Execution Time

We study different GKM clustering algorithms and examine their advantages and limitation. GKM provides solution to overcome the drawbacks of k-means algorithm but it has its own limitations like slow execution and large space requirement. To reduce these drawbacks of GKM number of solution and methods had been proposed which was efficient in comparison to GKM. We summarized most of them in our critical analysis section. Our algorithm requires less computing time and fewer distance calculations. It will also take low memory space.

VII. Conclusions

The proposed clustering methods are tested on well-known data sets and they compare favorably to the k-means algorithm with random restarts. In addition, the modified versions lead to implementations that are very fast and exhibit almost equal performance. We have presented the global k-means clustering algorithm, which constitutes a deterministic clustering method providing excellent results in terms of the clustering error criterion. The method is independent of any starting conditions and compares favorably to the k-means algorithm with multiple random restarts. The deterministic nature of the method is particularly important in cases where the clustering method is used either to specify initial parameter values for other methods (for example RBF training) or constitutes a module in a more complex system. In such a case we can be almost certain that the employment of the global k-means (or any of the fast variants) will always provide sensible clustering solutions. Therefore, one can evaluate the complex system and adjust critical system parameters without having to worry for dependence of system performance on the clustering method employed.

Another advantage of the proposed technique is that in order to solve the M-clustering problem, all intermediate k-clustering problems are also solved for k =1; : : : ; M. This may prove useful in many applications where we seek for the actual number of clusters and the k-clustering problem is solved for several values of k. We have also developed the fast global k-means algorithm, which signi2cantly reduces the required computational e3ort, while at the same time providing solutions of almost the same quality.

References

[1]. *Abney, S. (1996). Partial Parsing via Finite-State Cascades. In Proceedings of Workshop on Robust Parsing, 8th EuropeanSummerSchool in Logic, Language, and Information. Prague, Czech Republic: 8–15.*

[2]. *ACE (2004). Annotation Guidelines for Entity Detection and Tracking (EDT). http://www.ldc.upenn.edu/Projects/ACE/.*

[3]. *Adam, C. K., Ng, H. T., and Chieu, H. L. (2002). Bayesian Online Classifiers for Text Classification and Filtering. In Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval. Tampere, Finland, ACM Press, New York: 97–104.*

[4]. *Adams, T. L., Dullea, J., Barrett, T. M., and Grubin, H. (2001). "Technology Issues Regarding the Evolution to a SemanticWeb." ISAS-SCI 1: 316–322.*

[5]. *Aggarwal, C. C., Gates, S. C., and Yu, P. S. (1999). On the Merits of Building Categorization Systems by Supervised Clustering. In Proceedings of EDBT-00, 7th International Conference on Extending Database Technology. Konstanz, Germany, ACM Press, New York: 352–356.*

[6]. *Agrawal, R., Bayardo, R. J., and Srikant, R. (2000). Athena: Mining-based Interactive Management of Text Databases. In Proceedings of EDBT-00, 7th International Conference on Extending Database Technulogy. Konstanz, Germany, Springer-Verlag, Heidelberg: 365–379.*

[7]. *Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. In Proceedings of the ACM SIGMOD Conference on Management of Data.Washington, DC, ACM Press, New York: 207–216.*

[8]. *Agrawal, R., and Srikant, R. (1994). Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Databases (VLDB-94). Santiago, Chile, Morgan Kaufmann Publishers, San Francisco: 487–499.*

[9]. *Agrawal, R., and Srikant, R. (1995). Mining Sequential Patterns. In Proceedings of the 11th International Conference on Data Engineering. Taipei, Taiwan, IEEE Press, Los Alamitos, CA: 3–14.*

[10]. *Agrawal, R., and Srikant, R. (2001). On Integrating Catalogs. In Proceedings of WWW-01, 10th International Conference on theWorldWideWeb. HongKong,ACMPress, NewYork: 603–612.*

[11]. *Ahlberg, C., and Schneiderman, B. (1994). Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In Proceedings of the International Conference on Computer-Human Interaction. Boston, ACM Press, New York: 313–317.*

[12]. *Ahlberg, C., and Wistrand, E. (1995). IVEE: An Information Visualization and Exploration Environment. In Proceedings of Information Visualization '95 Symposium. Atlanta, GA, IEEE, Los Alamitos, CA: 66–73.*

[13]. *Aho, A., Hopcroft, J., and Ullman, J. (1983). Data Structures and Algorithms. Reading, MA, Addison-Wesley.*

[14]. *Ahonen-Myka, H. (1999). Finding AllFrequent Maximal Sets inText. In Proceedings of the 16th International Conference on Machine Learning, ICML-99Workshop on Machine Learning in Text Data Analysis. Ljubljana, AAAI Press, Menlo Park, CA: 1–9.*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

[15]. *Ahonen, H., Heinonen,O., Klemettinen, M., andVerkamo, A. (1997a). Applying Data Mining Techniques in Text Analysis. Helsinki, Department of Computer Science, University of Helsinki.*

[16]. *Ahonen, H., Heinonen, O., Klemettinen, M., and Verkamo, A. (1997b). Mining in the Phrasal Frontier. In Proceedings of Principles of Knowledge Discovery in Databases Conference. Trondheim, Norway, Springer-Verlag, London.*

[17]. *Aitken, J. S. (2002). Learning Information Extraction Rules: An Inductive Logic Programming Approach. In Proceedings of the 15th European Conference on Artificial Intelligence. Lyon, France, IOS Press, Amsterdam.*