

# Search Technique Using Wildcards or Truncation: A Tolerance Rough Set Clustering Approach

Sandeep Kumar Satapathy, Shruti Mishra

Department of Computer Science & Engineering,  
Institute of Technical Education & Research,  
Siksha O Anusandhan University,  
Bhubaneswar, Odisha, INDIA  
sandeepkumar04@gmail.com,  
shruti\_m2129@yahoo.co.in

Debahuti Mishra

Department of Computer Science & Engineering,  
Institute of Technical Education & Research,  
Siksha O Anusandhan University,  
Bhubaneswar, Odisha, INDIA  
debahuti@iter.ac.in

**Abstract-** Search engine technology plays an important role in web information retrieval. However, with Internet information explosion, traditional searching techniques cannot provide satisfactory result due to problems such as huge number of result Web pages, unintuitive ranking etc. Therefore, the reorganization and post-processing of Web search results have been extensively studied to help user effectively obtain useful information. This paper has basically three parts. First part is the review study on how the keyword is expanded through truncation or wildcards (which is a little known feature but one of the most powerful one) by using various symbols like \* or !. The primary goal in designing this is to restrict ourselves by just mentioning the keyword using the truncation or wildcard symbols rather than expanding the keyword into sentential form. The second part of this paper gives a brief idea about the tolerance rough set approach to clustering the search results. In tolerance rough set approach we use a tolerance factor considering which we cluster the information rich search result and discard the rest. But it may so happen that the discarded results do have some information which may not be up to the tolerance level; still they do contain some information regarding the query. The third part depicts a proposed algorithm based on the above two and thus solving the above mentioned problem that usually arise in the tolerance rough set approach. The main goal of this paper is to develop a search technique through which the information retrieval will be very fast, reducing the amount of extra labor needed on expanding the query.

**Keywords:** Clustering, Tolerance Rough Set, Search Engine, Wildcard Truncation

## I. INTRODUCTION

With rapid development of Internet technologies and Web explosion, searching useful information from huge amount of Web pages becomes an extremely difficult task. Currently Internet search engines are the most important tools for Web information acquisition. Based on techniques such as Web page content analysis, linkage analysis, etc., search engines locate a collection of related Web pages with relevance rankings according to user's query. However, current search results usually contain large amount of Web pages, or are with unintuitive rankings, which makes it inconvenient for users to find the information they need. Therefore, techniques for improving the organization and presentation of the search results have recently attracted a lot of research interest. The typical techniques for reorganizing search results include Web page clustering, document summarization, relevant information extraction, search result

visualization, etc. Wildcards are one of the searching techniques which are further improved to provide an effective way of searching according to the user's specification. One approach to manage large results set is by clustering. Tolerance Rough Set Model (TRSM) was developed [1,2] as basis to model documents and terms in information retrieval, text mining, etc. With its ability to deal with vagueness and fuzziness, tolerance rough set seems to be promising tool to model relations between terms and documents.

The earliest work on clustering results were done by Pedersen, Hearst et al. on Scather/Gather system [12], followed with application to web documents and search results by Zamir et al. [15,19] to create Grouper based on novel algorithm Suffix Tree Clustering. Inspired by their work, a Carrot framework was created to facilitate research on clustering search results. This has encouraged others to contribute new clustering algorithms under the Carrot framework like LINGO, AHC. Other clustering algorithms were proposed for, Semantic Hierarchical Online Clustering using Latent Semantic Indexing to cluster Chinese search results or Class Hierarchy Construction Algorithm by Schenker et al [20].

In this paper, we propose an algorithm based on the search results obtained by using wildcards or truncations and then applying the Tolerance Rough Set concept for clustering the search results. The rest of the paper is arranged like this, Section I gives the introductory concepts of Web acquisition concepts with the keyword searching with it's advantages and truncation mechanism, In section II, we present the abstracted view of the document clustering with it's definition, information retrieval, vector space model and other models. Section III focuses on Tolerance Rough Set Model, Section IV describes our proposed algorithm and finally section V depicts the conclusion.

### A. Keyword Searching

Keyword searching permits you to search a database for the occurrence of specific words or terms, regardless of where they may appear in the database record. For example, even if the word appears in the middle of the title of an article, or anywhere in the abstract, you can still search for it. Keyword searching was made possible by computers; essentially, the computer looks for any group of characters that has a space on either side of it, considers it a "word,"

and indexes it. The computer takes this task very literally. Even typos, ("philosophy"), incorrect spellings ("archaeology"), or words that were accidentally typed together without a space between them ("for example"), will be found by the computer and indexed, exactly the way they appear.

**B. Advantages of Keyword Searching**

There are many advantages to keyword searching [4, 5]: you can locate a very specific reference, even if it is only mentioned a single time. You can use the most current terminology, jargon, or "buzzwords" being used in a discipline, even when no official subject headings exist yet for the concept. You can combine keywords in various ways to create a very detailed and specific search query; the actual search as you enter it is known as a search statement. As you begin to search for information on your topic, develop a list of keywords and phrases that represent the most important aspects of your topic. Background information located in books and reference sources can be useful sources for these keywords. Try to come up with at least three words to describe each concept, grouping the keywords by concept. You can then use these keywords to "ask" the computer to search for the specific words and phrases on your list.

For example, if you were researching the effect of the media on body image and eating disorders, your keyword lists might look like this:

TABLE I: Keyword List

Concept #1	Concept#2	Concept#3
Media* Mass me \$ ia Television\$ TV Advertis! Film Movies	Body image Self-esteem	Eating Disorders Anorexia Bulimia

**C. Truncation**

Truncation [9, 10,11] allows you to search for alternate forms of words. Shorten the word to its root, then add a special character (\*, \$, !). When truncating, be sure to include enough of the search word to make it meaningful. For Example, if you wanted to search for alternative forms of the word advertising, a good choice would be to truncate it as "advertis." This will search words such as advertise, advertising, and advertisement. You wouldn't, however, want to truncate after the adv. If you did, your search would include words such as advantage, advance, adventure, advice, etc.

Different indexes and databases [3] use different symbols after the root word to accomplish truncation. If you are unsure of the truncation symbol for the database you are using, consult the help section for that resource. The most common truncation symbols are: \*, \$, !.

TABLE II: Truncation List

Symbol	Database	Example
*	<ul style="list-style-type: none"> <li>CONSORT/Ohio LINK</li> <li>Web of Science</li> <li>Yahoo</li> </ul>	advertis*

\$	<ul style="list-style-type: none"> <li>Periodical Abstracts</li> <li>Humanities Abstracts</li> <li>Biological Abstracts</li> <li>MLA Bibliography</li> </ul>	Advertis\$
!		Advertis!

**II. DOCUMENT CLUSTERING**

**A. General Definition**

Clustering is an established and widely known technique for grouping data. It has been recognized and found successful applications in various areas like data mining [6,7], statistics and information retrieval [1, 8].

Let  $D = \{d_1, d_2, d_3, \dots, d_n\}$  be a set of objects, and  $\delta(d_i, d_j)$  denote a similarity measure between objects  $d_i, d_j$ . Clustering then can be define as a task of finding the decomposition of  $D$  into  $K$  clusters  $C = \{c_1, c_2, \dots, c_k\}$  so that each object is assigned to a cluster and the ones belonging to the same cluster are similar to each other (regarding the similarity measure  $d$ ), while as dissimilar as possible to objects from other clusters.

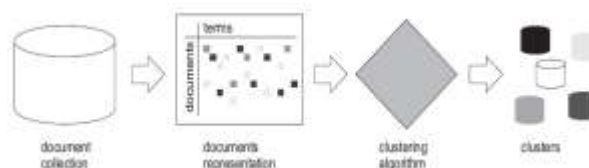


Figure 1. Document Clustering Process

There are numerous clustering algorithms ranging from vector-space based, model-based (mixture resolving) to graph-theoretic spectral approaches. However, when concerning application to text, algorithms based on vector space are the most frequently used. In this work we will concentrate on vector space and provide a detail analysis of vector-based algorithms for document clustering. A readers interested in other clustering approaches is referred to [6,7,10].

**B. Clustering in Information Retrieval**

In the figure 1 while clustering has been used in various task of Information Retrieval (IR) [11, 13], it can be noticed that there are two main research themes in document clustering: as a tool to improve retrieval performance and as a way to organizing large collection of documents. Document clustering for retrieval purposes originates from the *Cluster Hypothesis* [15] which states that *closely associated documents tend to be relevant to the same requests*. By grouping *similar* documents together, one hopes that relevant documents will be separated from irrelevant ones, thus performance of retrieval in the clustered space can be improved. The second trend represented by [6, 10] found clustering to be a useful tool when browsing large collection of documents. Recently, it has been used in [15, 16] for grouping results returned from web search engine into thematically related cluster.

Several aspects need to be considered when approaching document clustering.

**C. Vector Space Model and Document Representation**

While in some domain such as data mining, objects of interest are frequently given in the form of feature/attributes vector, documents are given as sequences of words. Therefore, to be able to perform document clustering, an appropriate representation for document is needed. The most popular method is to represent documents as vectors in multidimensional space. Each dimension is equivalent to a distinct term (word) in the document collection. Due to the nature of text documents, the number of distinct terms (words) can be extremely large, counting in thousands for a relatively small to medium text collection. Computation in that high-dimensional space is prohibitively expensive and sometimes even impossible (e.g. memory size restriction). It is also obvious that not all words in the document are equally useful in describing its contents. Therefore, documents needs to be preprocessed to determine most appropriate terms for describing document semantic - *index terms*.

Assume that there are  $N$  documents  $d_1, d_2, d_3, \dots, d_n$  and  $M$  index terms enumerated from 1 to  $M$ . A document in vector space is represented by a vector:

$$d_i = [w_{i1}, w_{i2}, \dots, w_{iM}] \quad \text{Equation (1)}$$

where  $w_{ij}$  is a weight for the  $j$ -th term in document  $d_i$ .

#### D. Term Frequency - Inverse Document Frequency Weighting

The most frequently used weighting scheme is TD\*IDF [2] (term frequency - inverse document frequency) and its variations. The rationale behind TD\*IDF is that terms that has high number of occurrences in a document (*tf factor*), are better characterization of document's semantic content than terms that occurs only a few times. However, terms that appears frequently in most documents in the collection will have little value in distinguishing document's content, thus the *idf factor* is used to downplay the role of terms that appears often in the whole collection.

In our work, we can construct a table containing the potential query refinement terms are selected from the top search results returned by the underlying Web search engine. However, rather than collecting the actual document contents, the frequency statistics are based only on the title and snippet provided by the underlying search engine. The title is often descriptive of the information within the document, and the snippet contains contextual information regarding the use of the query terms within the document. These both provide valuable information about the documents in the search results.

Let  $t_1, \dots, t_m$  denotes terms in the document corpus and  $d_1, \dots, d_n$  are documents in the corpus. In TD\*IDF, the weight for each term  $t_j$  in document  $d_i$  is defined [13] as

$$w_{ij} = tf_{ij} * \log(n / df_j) \quad \text{Equation (2)}$$

where  $tf_{ij}$  (term frequency, *tf*) - number of times term  $t_j$  occurs in document  $d_i$ ,  $df_j$  (document frequency) - number of documents in the corpus in which term  $t_j$  occurs. The factor  $\log(N/df_j)$  is called *inverse document frequency (idf)* of term.

### III. TOLERANCE ROUGH SET MODEL

Tolerance Rough Set Model (TRSM) was developed [17, 18, 19] as basis to model documents and terms in information retrieval, text mining, etc. With its ability to deal with vagueness and fuzziness, tolerance rough set seems to be promising tool to model relations between terms and documents. In many information retrieval problems, especially in document clustering, defining the relation (i.e. similarity or distance) between document-document, term-term or term-document is essential. In Vector Space Model, it has been noticed [18, 20] that a single document is usually represented by relatively few terms. This results in zero-valued similarities which decreases quality of clustering. The application of TRSM in document clustering was proposed as a way to enrich document and cluster representation with the hope of increasing clustering performance.

#### A. Tolerance Space of Terms

Let  $D = \{d_1, d_2, d_3, \dots, d_n\}$  be a set of document and  $T = \{t_1, t_2, \dots, t_m\}$  set of *index terms* for  $D$ . With the adoption of Vector Space Model each document  $d_i$  is represented by a weight vector  $[w_{i1}, w_{i2}, \dots, w_{iM}]$  where  $w_{ij}$  is a weight for the  $j$ -th term in document  $d_i$ . In TRSM, the tolerance space is defined over a universe of all index terms:

$$U = T = \{t_1, t_2, \dots, t_m\} \quad \text{Equation (3)}$$

The idea is to capture conceptually related index terms into classes. For this purpose, the tolerance relation  $R$  is determined as the co-occurrence of index terms in all documents from  $D$ . The choice of co-occurrence of index terms to define tolerance relation is motivated by its meaningful interpretation of the semantic relation in context of IR and its relatively simple and efficient computation.

#### B. Tolerance Class of Term

Let  $f_D(t_i, t_j)$  denotes the number of documents in  $D$  in which both terms  $t_i$  and  $t_j$  occurs. The uncertainty function  $I$  with regards to threshold  $\theta$  is defined as

$$I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\} \quad \text{Equation (4)}$$

Clearly, the above function satisfies conditions of being reflexive:  $t_i \in I_\theta(t_i)$  and symmetric:  $t_j \in I_\theta(t_i) \Leftrightarrow t_i \in I_\theta(t_j)$  for any  $t_i, t_j \in T$ . Thus, the tolerance relation  $I \subseteq T \times T$  can be defined by means of function  $I$ :

$$t_i I t_j \Leftrightarrow t_j \in I_\theta(t_i) \quad \text{Equation (5)}$$

where  $I_\theta(t_i)$  is the tolerance class of the index term  $t_i$ .

In context of Information Retrieval, a tolerance class represents a concept that is characterized by terms it contains. By varying the threshold  $\theta$  (e.g. relatively to the size of document collection), one can control the degree of relatedness of words in tolerance classes (or in other words the preciseness of the concept represented by a tolerance class).

To measure degree of inclusion of one set in another, vague inclusion function is defined as:

$$v(X, Y) = |X \cap Y| / |X| \quad \text{Equation (6)}$$

It is clear that this function is monotonous with respect to the second argument. The membership function  $\mu$  for  $t_i \in T, X \subseteq T$  is then defined as:

$$\mu(t_i, X) = v(I_{\theta}(t_i), X) = |I_{\theta}(t_i) \cap X| / |I_{\theta}(t_i)| \quad \text{Equation (7)}$$

With the assumption that the set of index terms  $T$  doesn't change in the application, all tolerance classes of terms are considered as structural subsets:  $P(I_{\theta}(t_i)) = 1$  for all  $t_i \in T$ .

Finally, the lower and upper approximations of any subset  $X \subseteq T$  can be determined with the obtained tolerance  $R = (T, I, v, P)$  respectively as:

$$L_R(X) = \{ t_i \in T \mid v(I_{\theta}(t_i), X) = 1 \} \quad \text{Equation (8)}$$

$$U_R(X) = \{ t_i \in T \mid v(I_{\theta}(t_i), X) > 0 \} \quad \text{Equation (9)}$$

One interpretation of the given approximations can be as follows: if we treat  $X$  as a concept described vaguely by index terms it contains, then  $U_R(X)$  is the set of concepts that share some semantic meanings with  $X$ , while  $L_R(X)$  is a "core" concept of  $X$ .

TITLE	Econ Papers: Rough sets bankruptcy prediction models versus auditor
DESCRIPTION	Rough sets bankruptcy prediction models versus auditor rates. Journal of Forecasting, 2003, vol.22, issue 8, pages 569-586. Thomas E. McKee.....

### C. Extended Weighting Scheme for Upper Approximation

To assign weight values for document's vector, the TF\*IDF weighting scheme is used. In order to employ approximations for document, the weighting scheme need to be extended to handle terms that occurs in document's upper approximation but not in the document itself (or terms that occurs in the document but not in document's lower approximation). The extended weighting scheme is defined as:

$$w_{ij} = \begin{cases} (1 + \log(f_{d_i}(t_j)) * \log \frac{N}{ID(t_j)}) & \text{if } t_j \in d_i \\ \min_{t_k \in d_i} w_{ik} * \frac{\log \frac{N}{ID(t_j)}}{1 + \log \frac{N}{ID(t_j)}} & \text{if } t_j \in U_R(d_i) \setminus d_i \\ 0 & \text{if } t_j \notin U_R(d_i) \end{cases} \quad \text{Equation (10)}$$

where  $w_{ij}$  is the weight for term  $t_j$  in document  $d_i$ .

The extension ensures that each terms occurring in upper approximation of  $d_i$  but not in  $d_i$ , has a weight smaller than the weight of any terms in  $d_i$ . Normalization by vector's length is then applied to all document vectors:

$$w_{ij} = \frac{w_{ij}}{\sqrt{\sum_{t_k \in d_i} (w_{ik})^2}} \quad \text{Equation (11)}$$

## IV. IV PROPOSED ALGORITHM

Our proposed algorithm works as follows:

- In the first step, the user gives the initial term or a long query using wildcard or truncation symbols (placing it anywhere in the term or query).
- Then in the second step the first 20 results are viewed and scanned thoroughly.
- The third step is to represent the result into a table. The term or the entire query occurring for highest number of times are calculated and are placed accordingly in the table.
- After displaying the table we would use the Tolerance Rough set approach and select the most appropriate or nearest search result and cluster them into one group according to the priority order.
- Then rather than discarding all the discarded search result we would again apply tolerance rough set approach to cluster them further as some more appropriate search result could be obtained. We can name it as "Rough search result".

### Step 1

Since this algorithm is applied in the post processing phase so any kind of Information Retrieval tool can be used. This returns a list of documents like Google or Yahoo.

### Step2

- The first 20 results are taken into consideration.
- In this step the search result produced can contain the whole term or a part of the term along with some other relevant terms for which we have used the symbols(\*, ?).

TABLE III: Weight of terms

Document vector			
Original		Using Upper Approximation	
Term	Weight	Term	Weight
auditor	0.567	auditor	0.564
bankruptcy	0.4218	bankruptcy	0.4196
signaling	0.2835	signaling	0.282
EconPapers	0.2835	EconPapers	0.282
rates	0.2835	rates	0.282
versus	0.223	versus	0.2218
issue	0.223	issue	0.2218
Journal	0.223	Journal	0.2218
MODEL	0.223	MODEL	0.2218
Prediction	0.1772	Prediction	0.1762
Vol	0.1709	Vol	0.1699
		applications	0.0809
		Computing	0.0643

### Step 3

- Now the weight of each data or term is calculated that has occurred for highest number of times.
- A table is formed having the frequency value along with the specific term with the type of data, which has occurred for the highest number of times.
- The table can contain highest frequency value first with the lowest term value at last or vice-versa.

### Step 4

- a. Here we use the Tolerance rough set approach. We consider a global similarity threshold or tolerance factor or level and determine the required level of similarity for inclusion within a tolerance class and the remaining search results are simply discarded. After that we can apply various clustering methodologies to cluster them into appropriate groups of different meanings.
- b. Once the table is displayed now it is up to the user to decide which particular or nearest data he/she is willing to view. A user can view the data by simply clicking on it.

### Step 5

Now here the main factor is taken into consideration. It may happen the discarded results do contain some meaningful information that the user might want to refer or have.

Here again use the tolerance rough set approach to the discarded results and again use a global threshold similarity function and cluster the appropriate results and name them as "Rough Search result".

These "Rough search results" are then displayed in a different section but in the same page where the Original tolerance set was displayed, so that a user can also have a quick reference to get some or other needed information.

## V. CONCLUSION

This paper has presented an interactive method for term or query Expansion using wildcards or truncation searching techniques (\*,!, \$) based on term weighting, tolerance rough set model and later clustering the roughness found. The method is found on the fact that documents contain some terms with high information content, which can summarize their subject matter. Those terms can be found out efficiently through this proposed algorithm. This particular algorithm helps us to save much some useful information that we generally omit during rough set analysis. But each day is passing and new advancements are coming into light. So, our future aspects would be to implement this strategy and make it more efficient to deal with. Also, our target would be to implement this strategy into various fields and industry to see how efficiently it works and also comparing it with other searching techniques so that we can make this as one of the best searching technique ever used till date.

## REFERENCES

- [1] S.Kawasaki, N.B.Nguyen, "Hierarchical document clustering based on tolerance rough set model", In *Principles of Data Mining and Knowledge Discovery, 4th European Conference, PKDD*, Lyon, France, 2000.
- [2] S.Mishra, S. K. Satapathy, D.Mishra, "Improved search technique using Wildcards or Truncation", In *the Proc. of Intelligent Agent & Multi-Agent systems*, Published at *IEEE Explore 978-1-4244-4711-4/09/*, 2009.
- [3] C.Silverstein, M. Hen zinger, H.Marais, M. Moricz, "Analysis of a very large AltaVista query log", In *the Proc. of Tech. Rep. 2008-014, Digital SRC*, 2008.
- [4] Google-Information literacy tutorial- Searching Techniques
- [5] Google-Duke Libraries-Searching Techniques.
- [6] J .Han, "Data Mining: Concepts and Techniques", 1st ed. Morgan Kaufmann, 2000.
- [7] B.Larsen, C.Aone, "Fast and erective text mining using linear-time document clustering", In *Proceedings of the fifth ACM SIGKDD International conference on Knowledge discovery and data mining*, pp. 16(22), 1999.

- [8] A. K.Jain, M. N. Murty, P. J.Flynn, "Data clustering: a review", In *the Proc. of ACM Computer. Survey. 31*, 3, 264(323), 2009.
- [9] R.Baeza-Yates, B.Ribeiro-Neto, "Modern Information Retrieval", 1st ed. Addison Wesley Longman Publishing Co. Inc., May 2007.
- [10] S.Chakrabarti, "Mining the Web", Morgan Kaufmann, 2003.
- [11] D.R.Cutting, D.R.Karger, J.O. Pedersen, J.W.Tukey, "Scatter/gather: a cluster-based approach to browsing large document collections", In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 318(329), 1992.
- [12] M. A.Hearst, J. O.Pedersen, "Reexamining the cluster hypothesis: Scatter/gather on retrieval results", In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, Zürich, CH, pp. 76(84), 2007.
- [13] T. Honkela, S.Kaski, Kohonen, T.Websom, "Self-organizing maps of document collections", In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6 1997, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland*. pp. 310 (315), 2007.
- [14] V. Rijsbergen, "Information Retrieval", In *the Proc. ACM Digital London, 2007*.
- [15] O.Zamir, O. Etzioni, "Grouper: a dynamic clustering interface to web search results", In *the proc. of the Computer Networks, Amsterdam, Netherlands, 31(11)*, 2008.
- [16] Chen Wu, J. Dai, "Information Retrieval Based on Decompositions of Rough Sets Referring to Fuzzy Tolerance Relations", In *the Proc. of IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.3, March 2009*.
- [17] R.Slowinski, D.Vanderpooten, "Similarity Relation as a Basis for Rough Approximations", In *Proceedings of the Second Annual Joint Conference on Information Science, Wrightsville Beach, N Carolina, USA, also ICS Research Report 53,1995:249-250*, 2009
- [18] Tu Bao Ho, N. B. N. "Nonhierarchical document clustering based on a tolerance rough set model", In *the Proc. of International Journal of Intelligent Systems 17, 2* 199(212), 2002.
- [19] O.Zamir, O.Etzioni, "Web document clustering: A feasibility demonstration", In *Research and Development in Information Retrieval* (1998), pp. 46-54.
- [20] A.Schenker, M.Last, "Design and implementation of a web mining system for organizing search engine results", In *Data Integration over the Web (DIWeb), First International Workshop, Interlaken, Switzerland, 4 June 2001*. (2001), pp. 62-75.

## AUTHORS PROFILE

Sandeep Kumar Satapathy is Asst.Prof. in the department of Computer Sc. & Engg. Institute of Technical Education & Research (ITER) under Siksha `O` Anusandhan University, Bhubaneswar. He has received his Masters degree from Siksha O Anusandhan University, Bhubaneswar, Odisha, INDIA. His research areas include Web Mining, Data mining etc.

Shruti Mishra is a scholar of M.Tech(CSE) Institute of Technical Education & Research (ITER) under Siksha O Anusandhan University, Bhubaneswar, Odisha, INDIA. Her research areas include Data mining, Parallel Algorithms etc.

Debahuti Mishra is an Assistant Professor and research scholar in the department of Computer Sc. & Engg, Institute of Technical Education & Research (ITER) under Siksha O Anusandhan University, Bhubaneswar, Odisha, INDIA. She received her Masters degree from KIIT University, Bhubaneswar. Her research areas include Datamining, Bio-informatics, Software Engineering, Soft computing. Many publications are there to her credit in many International and National level journal and proceedings. She is member of OITS, IAENG and AICSIT. She is an author of a book Aotumata Theory and Computation by Sun India Publication (2008).