ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

# An Approach towards Forming a Domain Dictionary to Ease the Process of Searching for Scholarly Publications

Aditya Kalivarapu[1], M Kranthi Kiran[2]

M. Tech Student, Dept. of C.S.E, ANITS, Visakhapatnam, India[1]

Assistant Professor, Dept. of C.S.E, ANITS, Visakhapatnam, India[2]

**ABSTRACT:** Now a day's a large number of scholarly publications are being published and most of the information saved in text form. Hence text mining has become an increasingly popular and also important field in the research of data mining. In the text mining one of the concepts is to find domain related articles. Retrieving domain related documents can be performed by searching of domain keywords in the documents. For performing the searching process, in this paper we are implementing fast query pattern matching algorithm. Using fast query pattern matching algorithm we can identify the domain related keywords in the document and also get related documents over the domain. After completion of getting domain related documents we can provide ranking for those documents. For performing this process we are implementing term frequency inverse document frequency weight schema. By performing those two operations we can get most domain related documents with a ranking highest priority. So that by implementing these concepts we can provide efficient searching result and also pro vide best ranking of individual domain related documents.

**KEYWORDS:** Data mining, term frequency inverse document frequency weight method (tf–idf), pattern matching

## I. INTRODUCTION

In computer science, text mining has become an important research area. The process of getting very useful information from an unstructured text is known as text mining. Text mining is same as data mining, except the tools designed for data mining are used to handle structured data from databases, but text mining can work with semi-structured data, whereas HTML files, emails, and full-text documents, etc, and also includes unstructured text. Information extraction is one of the technologies that have been developed and can be used in the text mining process. In text mining, information extraction is an important research area. One of the text mining techniques is information extraction which means extracting structured information from unstructured documents and semi-structured documents.
The fast query pattern matching algorithm proposed in this paper is used for retrieving relevant scholar publications based on the query string given by the user and a domain dictionary is maintained. A domain dictionary in the research field refers to a collection of most used keywords in a particular domain. Basing on the keywords present in a document its domain can be recognized. Term Frequency-Inverse Document Frequency Weight Method is used for providing ranking for the retrieved documents.

## II. LITERATURESURVEY

In [11] Yves Rasolofo et al. proposed a distributed information retrieval framework based on the Okapi probabilistic model, a framework capable of achieving the same levels of retrieval effectiveness as those achieved by a single centralized index system. Here, the impact of a new term proximity algorithm on retrieval effectiveness for a keyword-based system was examined. It improves ranking for documents having query term pairs occurring within a given distance constraint. It potentially improves the precision after retrieving a few documents and thus could prove useful for those users looking only at the top ranked items. In [12] Sanjay Agarwal et al. proposed DBXplorer: A System for Keyword-Based Search over Relational Databases; which is implemented using a commercial relational database and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

web server and allows users to interact via a browser front-end.While traditional database management systems offer powerful query languages, they do not allow keyword-based search. In [13] Shermann loyal men et al. has proposed a System and method for context-based document retrieval. It addresses a major problem in text-based document retrieval: rapidly finding a small subset of documents in a large document collectionthat are relevant to a limited set of query terms supplied by the user. It is based on utilizing information contained in the document collection about the statistics of word relationships to facilitate the specification of search queries and document comparison. It retrieves documents with contextual similarity rather than word frequency similarity, simplifying search specification while allowing greater search precision.

## III. PROPOSED SYSTEM

In this paper we are proposed an efficient process for searching the domain related electronic document and provide ranking for those documents. Before perform the searching process the administrator will upload the each domain related documents into server. After that the user will enter query for searching the documents. Take the query string as input of algorithm and perform fast query pattern matching algorithm for retrieving query related domain documents. Before retrieving domain related documents we can also perform the ranking process for finding highest query matched domain documents. The implementation procedure of fast query pattern matching algorithm is as follows.

**Fast Query Pattern Matching Algorithm:**

In this paper we are proposing a new pattern matching algorithm on the basis of window, called fast query pattern matching algorithm. Each match starts from the Search outset position of each window, and create a new structure of the algorithm. After having matched the Search outset position, scan the prefix of the pattern from beginning of the pattern, if matched fully, and then scan the suffix of the pattern from end of the pattern. This will be able to make full use of the nature of the pattern. In the fast query pattern matching algorithm we are generating ASCII shifted table.

### 1. ASCII Shifted Table:

The ASCII shifted table is generated as follows.

### a. Handling Alphabets:

In the ASCII shifted table we are use size of alpha bet is to size (0-255) and then the size is first level is to size. In the ASCII shifted table each character user its decimal base value corresponding with its ASCII value of the position.

### b. Handling of query pattern:

In the query pattern we take each character in the patter from left to right and then give the position of each character which appears in the query pattern string in decreasing order. Take the position of each character which is indicated with the ASCII which would constitute a chain of other levels.

### c. Marking characters in query pattern string or not:

Take each character from the pattern and scan the ASCII shifted table, if the character occurs its corresponding position is incremented by one. If the character does not exists in the pattern then put zero in its corresponding position. By implementing this process we can know whether a character occurred in the pattern.

### 2. Search Starting position:

Take the text from the scholar article and defined special position is as starting of the text. Here we are taken search starting position as a center and take m-1 characters in itspattern. So that each compose of string contains 2m-1 characters

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

where m is length of string. In the total length of string we consider m-1 as first half of the characters and next m-1 window is the remaining half of the characters. By implementing this algorithm guarantees that the searching will improve accuracy.

**3. Next Array:**

In the searching process we are not found any match in the string we can go to back by using the next array. The next array will contain their own characteristics and nothing to do with the text string. In advance we are taken patterns P= $p_1,p_2,p_3$……..$p_m$ and generated function is next[i] within the range of 0<i<m+1. Suppose that in the searching process we are not finding any match of ith position we can calculate prefix of $p_1,p_2,p_3$ ….$p_{i-1}$ whether there is maximum of G making $p_1$……..$p_{g-1}$ = $p_{g-i+1}$ …$p_{i-1}$ . If pattern exists in the string then go to next[i] =G and pattern directly moved backward for i-next[i]. After performing backward operation then start the comparison from $G^{th}$ of pattern string if it not exists there will be next[i]=1.

**4. Matching of pattern in string:**

In the text string each match will start search from starting position and it also use ASCII shifted table, next array.

1. First examine the ASCII shifted table of search starting position is whether 1 or 0. If it occurs 1 then character occurs in the pattern of string otherwise go to next character of string.
In the second level of ASCII shifted table find the first position of character in the pattern and balance the string pattern in location of first position of character with the text string in a position to $k^{th}$ search starting position.

2. If you find the match from left side of the pattern then complete search starting position and go to match from right side of the pattern. After completing both sides searching process it will find matched pattern and go to next search staring position of text string.

3. Suppose that if a match in certain position fails then checkASCII shifted table and find the next position of character in the $k^{th}$ search starting position. After finding next search starting position calculate distance between two position and compare distance with next[i] size. By calculating distance take the jump of finding position of string of characters of pattern.

4. If starting position matches with pattern then matching process is done as in step 2. If the pattern is not matched go to step 3. This process will repeat until completion of text string.

After completion of finding domain related keywords in the documents we will give ranking for documents. For providing ranking of each document, term frequency inverse document frequency weight schema is used. The implementation process of term frequency inverse document frequency weight schema is as follows.

## Term Frequency-Inverse Document Frequency Weight Schema:

In this module we are provide ranking of documents in a particular domain. Before providing ranking of documents we can identify which document related to specified domain. By using fast query pattern matching algorithm we can identify which document are related to specified domain. After completion of retrieving domain related document we can provide ranking. Following steps are used to finding ranking of documents in a specified domain.

1. Term frequency inverse document frequency weight schema is statistical domain that evaluates weight of each domain keywords in documents. Before calculating document weight we can calculate term frequency of specified domain keywords.

$$TF = nt,i/ \sum i\text{-}1\ k\ nk,i$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

nt,i=number occurrence of Domain keywords in the document.
nk,I =number of occurrence of all term in a document.

2. After calculation of term frequency we can find out inverse document frequency. We can calculate inverse document frequency by using following formula.

Idf=log M/mt + 0.01

Where M= total number of documents in domain
mt = total number of document in domain where word t occurs.

3. The completion of inverse document frequency we can calculate weight of term in the document d by using following formula.

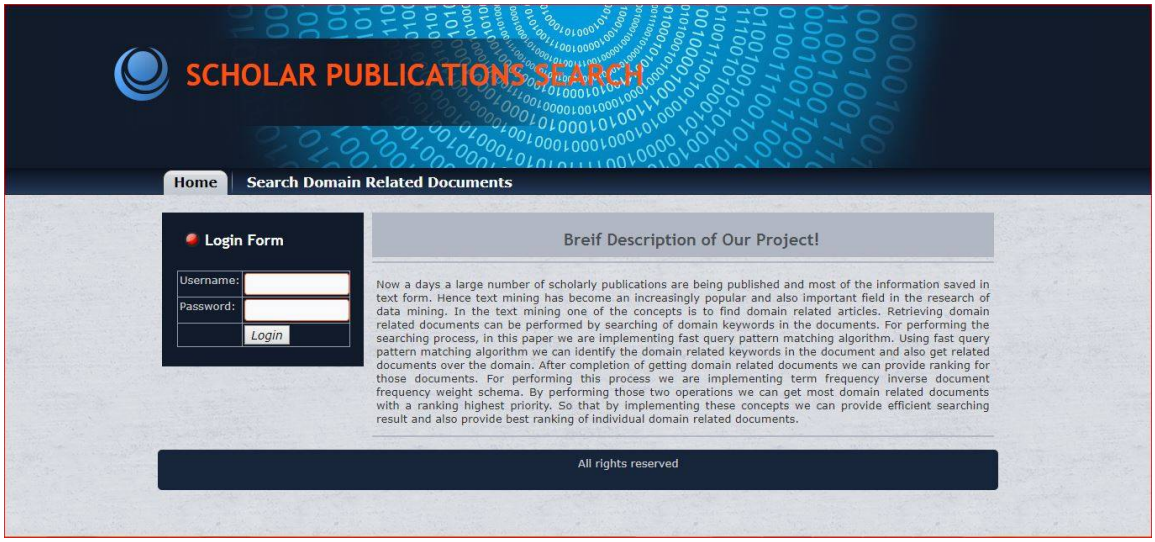
W(t, d)=TF(T,di)* idft

4. After calculating each document weight those documents will arranged in a descending order. By arranging those documents we can get highest matched domain keywords in first ranking and remaining also follows the same order.

By implementing those concepts we can get domain related documents and also provide ranking of each document in a domain.

## IV. RESULTS AND OBSERVATIONS

The result obtained consists of relevant documents from the domain matched with the given query string. The retrieved documents are listed in the order such that the document which is most relevant to the query string is retrieved first. Along with the retrieved document a link is provided for downloading the required document by the user.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

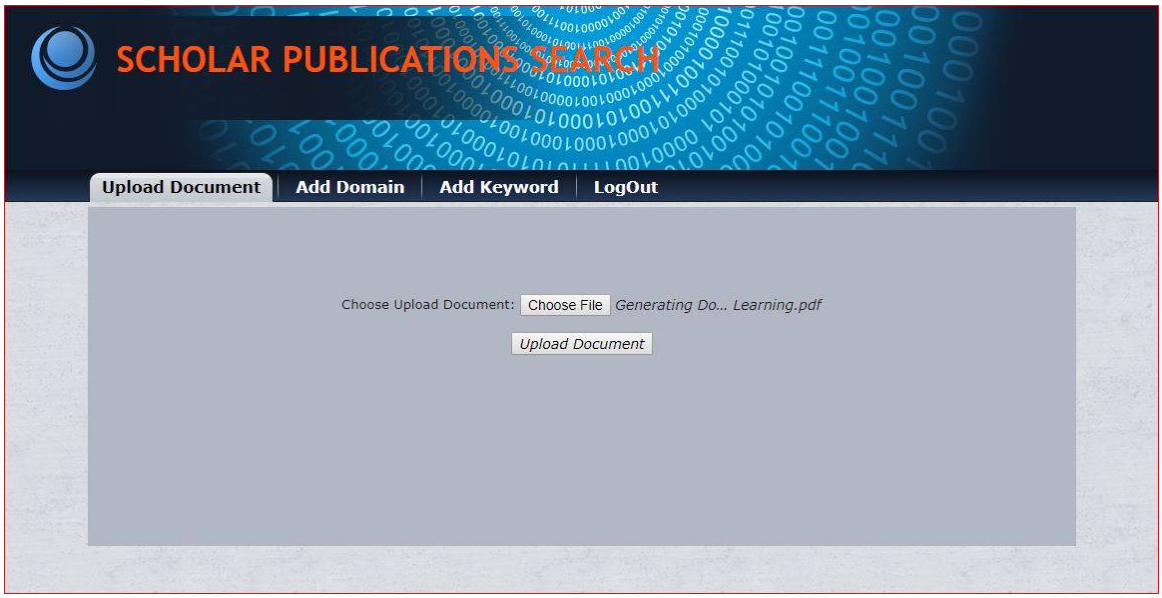
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

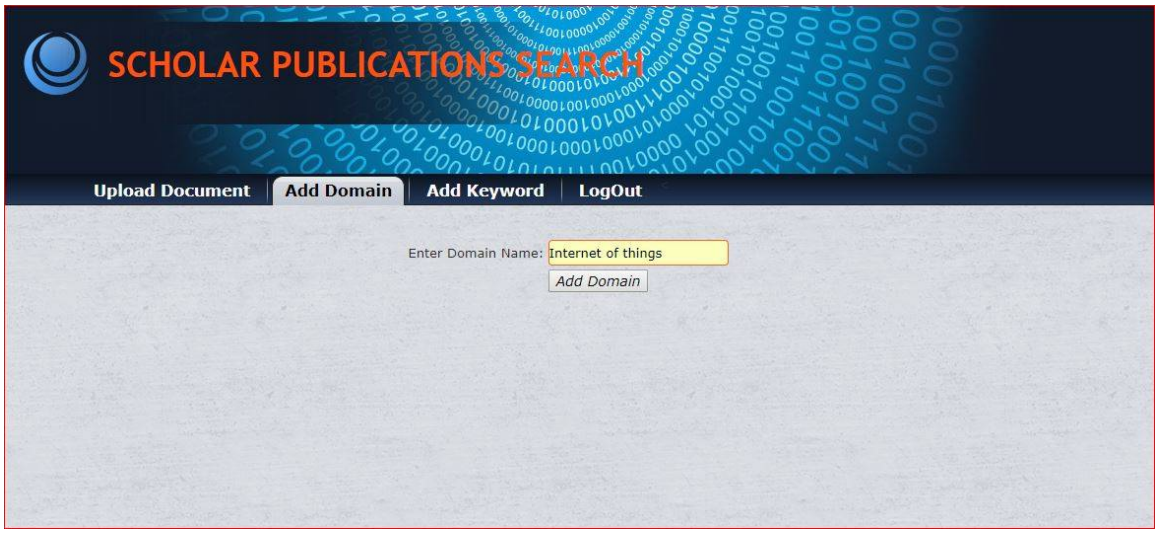
Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

**STEP 1: The Home page consists of two modules i.e admin and user modules. The admin logins into the system with required credentials. On logging into the system admin can upload documents, add domains and add keywords to the database.**




**Step 2:Uploading documents- the admin uploads all kinds research documents irrespective of their domain into the database**




**Step 3: Adding domain names- a domain refers to the research area, the domain names are added into the database. On successfully adding a domain name a text file is created which is used for storing the related keywords.**

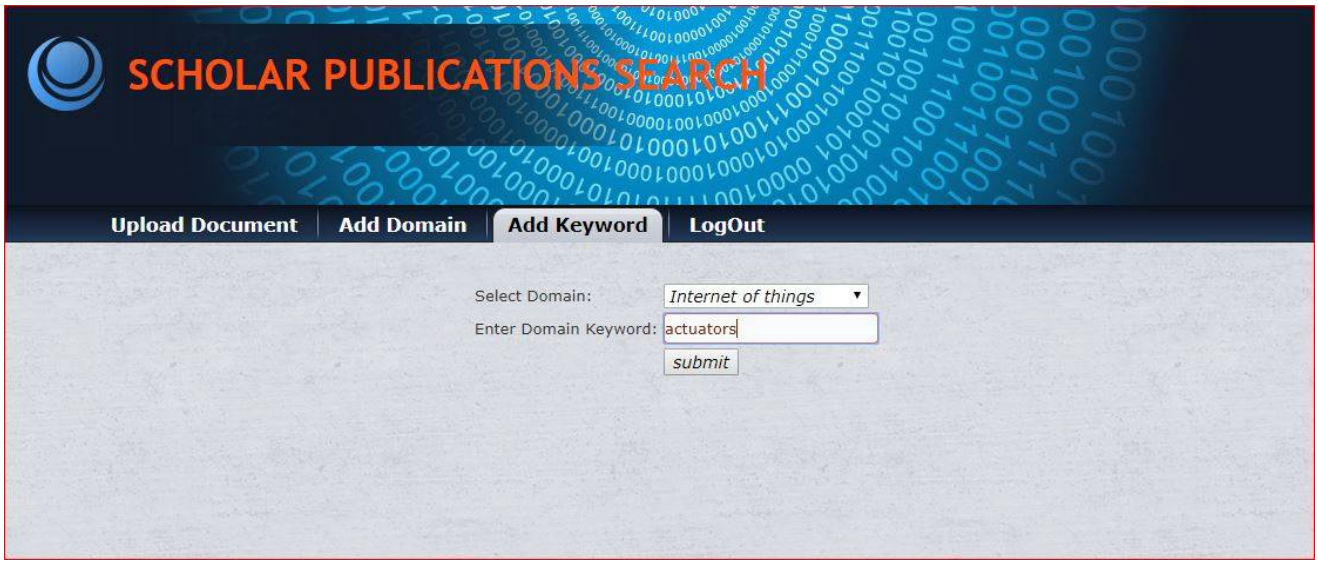
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

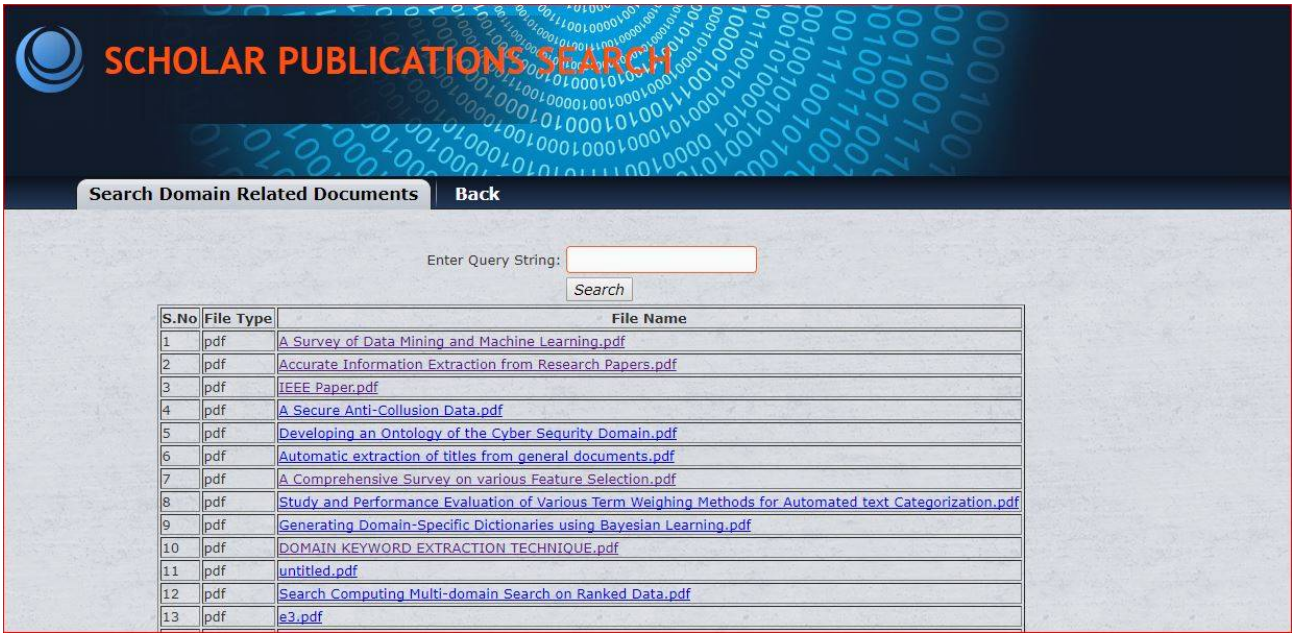
(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017




**Step 4: Adding keywords- keywords are added with respect to the specific domains into the database which are used for matching with the query string. The keywords are stored in the database in their respective domains.**




**Step 5: Retrieving documents: when a text is given as query string it performs the search operation by matching with the keywords present in the respective domain and retrieve the relevant documents based on the ranking.**

## IV. CONCLUSION

The two staged process presented in this paper is used for retrieving relevant documents and providing ranking based on keyword used to search. By adding the keywords to the relevant domains the domain dictionary is also maintained. The future work is to make the construction of domain dictionary automatic so that the keywords list for each domain gets updated by extracting information from the documents.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

## REFERENCES

1. Fukumoto, F., Suzukit, Y., & Fukumoto, J. I. (1997, March). An automatic extraction of key paragraphs based on context dependency. In Proceedings of the fifth conference on Applied natural language processing (pp. 291-298). Association for Computational Linguistics. J
2. Harish, B. S., and M. B. Revanasiddappa. "A Comprehensive Survey on various Feature Selection Methods to Categorize Text Documents." International Journal of Computer Applications 164.8 (2017).
3. Peng, F., and A. McCallum. "Accurate information extraction from research papers using conditional random fields. Retrieved on April 13, 2013." N04-1042.
4. Riloff, Ellen. "An empirical study of automated dictionary construction for information extraction in three domains." Artificial intelligence 85.1 (1996): 101-134.
5. Chakraborty, Rakhi. "DOMAIN KEYWORD EXTRACTION TECHNIQUE: ANew WEIGHTING METHOD." Computer Science & Information Technology109 (2013).
6. Nagao, Makoto, Mikio Mizutani, and Hiroyuki Ikeda. "An Automatic Method of the Extraction of Important Words from Japanese Scientific Documents." IPS Japan 17.2 (1976).
7. Debole, Franca, and Fabrizio Sebastiani. "Supervised term weighting for automated text categorization." Text mining and its applications. Springer, Berlin, Heidelberg, 2004. 81-97.
8. Aho, Alfred V., and Margaret J. Corasick. "Efficient string matching: an aid to bibliographic search." Communications of the ACM 18.6 (1975): 333-340.
9. Alfred, V. "Algorithms for finding patterns in strings." Algorithms and Complexity 1 (2014): 255-300.
10. Velardi, Paola, Paolo Fabriani, and Michele Missikoff. "Using text processing techniques to automatically enrich a domain ontology." Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. ACM, 2001.
11. Yves Rasolofo and Jacques Savoy.”Term Proximity Scoring for Keyword-Based Retrieval Systems” Published in Lecture Notes in Computer Science 2633, 1611-3349, 2003
12. S. Agrawal S. Chaudri G. Das “DBXplorer: a system for keyword based search over relational databases ”Proceedings of the 18th International Conference on Data Engineering (ICDEí02)1063-6382/02 $17.00 © 2002 IEEE.
13. Shermann Loyall Min, Constantin Lorenzo Tanno, Zachary Frank Mainen, William Russell Softky,System and method for context-based document retrieval US 6633868 B1x

## BIOGRAPHY

Aditya K is a M. Tech student in the department of computer science and engineering, ANITS, Visakhapatnam. His research interests are data mining, big data analytics.

M Kranthi Kiran is a assistant professor in the department of computer science and engineering, ANITS, Visakhapatnam. He is life member of CSI. He published several papers in international journals. He has 12 years of experience in teaching and research. His research interest is software engineering, sematic technologies.