"Bodh", BPIT's International Journal of Technology & Management
ISSN: 2454-8421, Volume 1, Issue 1, July-Dec, 2015, *Page 44-49*

Database optimization and Novelty Mining of News articles

Shweta Taneja, Charu Gupta, Ankita Mohan Saxena ,Jatin Rijhwani ,Sanya Malhotra

Department of Computer Science and Engineering, Bhagwan Parshuram Institute of Technology, Rohini, Delhi
shweta_taneja08@yahoo.co.in, charu_2287@yahoo.com

***Abstract*— With rapid advances in Information Technology, the normal way for people to obtain information has changed. However, the current available search engines, like Google, cannot tell whether a newly posted article contains fresh content or not, as compared to all the previous posted articles. Thus, people may sometimes waste time reading articles which are about old or have known information. The solution to this problem is Novelty mining; it is a new and exciting area of computer science research that tries to solve the crisis of information overload by combining techniques from data mining, text mining, natural language processing, information retrieval, and knowledge management.. In this paper, we have illustrated the various steps involved in the miming of the dataset. The objective of this research was to find an optimal way to scan through large databases and detecting relevant information efficiently. The results show that proposed novelty mining framework can detect novelty on a set of news articles with very high accuracy.**

***Key words* — Novelty Mining; Database optimization; Preprocessing; Information retrieval; Indexing**

I. Introduction

In today's information age, it is easy to store large amounts of data. However, although the amount of data available to us is continuously growing, our ability to gather this information and use it remains constant. Imagine the time savings if we are only presented with novel information to read, while the old or redundant information is filtered out. Thus, novelty mining [1] helps to extract novel information out of a huge set of text documents. The term novelty (derived from Latin word Novus for "new") is the quality of being new, or following from that, of being striking, original or unusual. In novelty mining, users are able to send different documents to be tested for its relevance and novelty. Due to the millions of data in the database, the insertion and selection of data have to be kept at optimum.

A novelty mining system [2] is able to discover novel, yet relevant information based on context and reader's preference. It is helpful in personal newsfeeds, information filtering, as well as many other fields where duplicate information may be returned to the users. In general, a novelty mining system consists of three main parts, namely preprocessing, classification and novelty mining. Firstly, text documents are input into the system for preprocessing, where models will be built by using various machine learning algorithms. Then, the system will determine relevant documents for a given topic and filter out the non-relevant documents in the classification stage. Finally, based on historical articles, the system will determine whether the input article is novel or not. The contributions of this paper are thus twofold. Firstly, to design and develop the optimizing techniques for SQL SERVER 2005 database for retrieval of relevant information, which has not been well-studied before and secondly, to study the novelty mining system which involves pre- processing as its first phase followed by classification and novelty mining techniques to detect novel data from a dataset. This paper spans across the three major emerging research areas of databases that include database indexing and information retrieval by query processing, pre-processing of dataset and knowledge management.

This paper is organized as follows. In the first section, brief introduction about the motivations for the research and development of novelty mining system is presented. In the second section literature review of various optimization and novelty mining systems is described. The third section comprises of the framework that we have proposed for our entire Novelty Mining system. In sections four details about the dataset used i.e. Reuters 21578 is explained. In section five and six the experiments conducted and subsequent performance evaluation is shown. Finally, at the end of this paper we conclude and give suggestions for future work in this field.

II. Related work

The major contribution in the field of optimization and novelty mining is by Flora S. Tsai. Other authors have also contributed in this area. In [1], authors have explored the importance of novelty mining and database optimization technique on a dataset of business blogs, with a very high accuracy. Previous research on novelty detection have stressed on the task of finding novel material, given a set of documents on a certain topic. Authors in [2] studied the part task defined by TREC 2002 novelty track that is firstly, finding the relevant sentences from the documents and then identifying the novel sentences from the collection of relevant ones. The research here shows that the former step appears to be more difficult part of the task. In [3], authors have analysed web logs posts for various categories of cyber security threats related to detection of cyber attacks, cyber crime and terrorism. They have used Latent Semantic models such as Latent Semantic Analysis (LSA) and Probabilistic LSA, to detect keywords from cyber security web logs. LSA is also discussed in another paper [6]. In another work [5], authors have proposed experimental results on APWSJ data set. They have shown that Document to Sentence(D2S) framework outperforms standard (document level) novelty detection in terms of redundancy-precision (RP) and redundancy-recall (RR). However they have suggested that D2S shows a strong capability to detect redundant information. Also, in [8] authors aim to explore the performance of redundancy and novelty mining in the business domain. They have adopted the mixed metric approach which combines symmetric and asymmetric metrics.

Different researchers have contributed in the area of database optimisation, but either they have focused on B–Trees or indexing techniques by LSA method. None has given attention to pre processing and optimisation using indexes. In our paper, we have proposed a framework which converts unstructured data of news articles to a structured form (tables) and there after indexing is performed and performance comparison is observed. This will also form basis for our future work of novelty mining, keeping in mind the constraints and challenges in natural text.

III. PROPOSED FRAMEWORK

The framework of Novelty Mining system is shown in figure 1. It is divided into four phases:-i. Pre-processing

ii. Database Optimization iii. Novelty Mining iv. Information Retrieval. The detailed explanation of these phases is given below.

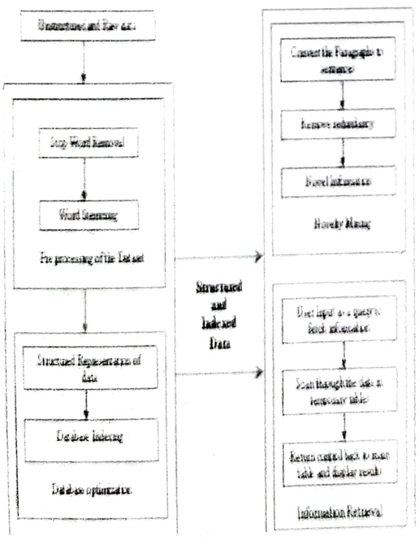

Unstructured and Raw data
Stop Word Removal
Word Stemming
Pre processing of the Data set
Structured Representation of data
Database Indexing
Database optimization
Structured and Indexed Data
Convert the Paragraphs to sentence
Remove redundancy
Novel Information
Novelty Mining
User input as a query to fetch information
Scan through the data in temporary table
Return control back to main table and display results
Information Retrieval

Fig. 1 . Proposed Framework

A. PRE- PROCESSING

There are various pre-processing techniques that infer or extract structured representations from raw unstructured data sources. There are different operations under pre processing like stop word removal and word stemming. Stop Word Removal aims to remove stop words like 'is', 'an', 'the' etc. Word Stemming is the process of reducing inflected (or sometimes derived) words to their stem, basic root form-generally a written word form. E.g. running-> run, Drinks-> drink, Mangoes-> mango.

1.) ALGORITHM USED FOR WORD STEMMING

We have used a modified form of Porter Stemmer Algorithm [10]. The Porter stemming algorithm (or 'Porter stemmer') is a process for performing stemming i.e., reducing the word to its root form. It is mainly used as a part of term pre-processing, that is usually done when setting up Information Retrieval systems. The algorithm stems the data using a set of rules. There are 60 rules in 6 steps of porter stemmer algorithm. These steps are:-

1. Removes plurals of the words.
2. Turns terminal y to i when there is another vowel in the stem.
3. Maps double suffixes to single ones, eg-'ization' , 'ational' etc
4. Deals with suffixes –full, -nests etc.
5. Takes off –ant, -ence etc.
6. Removes a final –e.

In our modified porter stemmer algorithm, we remove stop words like 'is', 'an', 'the' etc along with above suffix removal. We have used java as a programming language for implementing our algorithm. The benefit of implementing porter stemmer is to enhance search process in the large pool of data and moreover to increase the efficiency of the entire system.

B. DATABASE OPTIMIZATION

Database optimization is a technique to improve the query performance with indexing and statistics. It can be defined as the optimization of resources used to increase throughput and minimize contention, enabling the largest possible CPU workload to be processed. In our paper we have used indexing to optimize the dataset.

There are two types of indexes that have been built on the data namely clustered index and non clustered index. The two types of indexes are as explained below:

1) NON-CLUSTERED INDEX

The data is present in random order [12], but the index specifies the logical ordering. The index keys are in sorted order, with the pointer to the record contained in the leaves of the tree. There can be more than one non-clustered index on a database table. Non-Clustered indexes have structures that are different from the data rows. A non clustered index key value o points to data rows that contain the key value. This is called as row locator. Its structure is determined on the basis of the type of storage of the data pages. A heap table [11] by definition is a table that doesn't have any clustered indexes.

Another case arises when no index is defined for a table at all. In that case the address of the first IAM page of the heap table itself is stored in the sysindexes table with indid = 0 as shown in figure 3. So, the full form of IAM is a little misleading; it would be better called as SAM (Storage Allocation Map or Space Allocation Map).

2) Clustered Index

Clustering modifies the data block [12] into a certain specific order to match the index. Therefore, only one clustered index can be created on a given database table. Clustered indices greatly increases the overall speed of retrieval, but usually only if the data is accessed sequentially in the same or reverse order of the clustered index.

Fewer data block reads are required as the physical records are in the sort order on disk, the next row item in the sequence is immediately before or after the last one, and so on. Some databases separate the data and index blocks into separate files, others put two completely different data blocks within the same physical file(s). An object is created where the physical order of rows is the same as the index order of the rows and the bottom (leaf) level of clustered index contains the actual data rows.

C. Novelty Mining

Novelty mining is the identification of new or unknown information from a given set of text documents. It is useful in personal newsfeeds, information filtering, as well as many other fields where duplicate information may be returned to the users.

The approach followed to perform Novelty mining in our project is Document to sentence. The document (or paragraph) of information is fragmented into sentences to remove duplicity or redundancy. Temporary tables are created on the fly to accomplish the task. Moreover the search is also carried out in these tables for patterns as specified by the user, but the information is displayed from the main table.

A Cursor makes it possible to perform complex logic in SQL. A cursor can be viewed as a pointer to a row. It can only refer one row at a time.

Two Cursors have been used in this project for the following purposes:

(1) For removing redundancy/duplicity.

(2) For implementing search in the temporary tables.

Cursor is a server side tool. It is giving row-wise solution to the result set.

D. INFORMATION RETRIEVAL

Retrieval of information is also an integral part while designing a system, so as to provide the user relevant information according to the query input by him. If the results obtained are relevant and correct, then the system developed is said to be efficient. The various steps for retrieval of information from the structured tables are as follows:

(1) User inputs the request through query.

(2) Search is carried out in temporary tables through pattern matching.

(3) Transfer of control from temporary tables to the main table.

(4) Output generated from the main table.

IV. DATASET

A. Reuters 21578

We have used Reuters 21578 dataset [9] in our work. The documents in the Reuters-21578 collection appeared on the Reuters newswire in 1987. We have used a subset of this complete dataset for our study. The dataset is divided into six categories namely companies, exchanges, organizations, people, exchanges and topics.

B. Tool Used

The database software that we have used is MICROSOFT SQL SERVER 2005.Microsoft SQL Server 2005 is a relational database management system developed by Microsoft. SQL Server 2005 (formerly codenamed "Yukon") was released in October 2005. It included native support for managing XML data, in addition to relational data.

V. Experiments Conducted

The various phases under the project such as pre processing, database optimization and novelty mining were carried out as under. Performance evaluation is a key step to examine a project. We have evaluated our work and calculated the efficiency of our work. But before discussing the various cases of execution, we first give an overview of the work done and then its relevant efficiency.

Step 1: Creation of tables

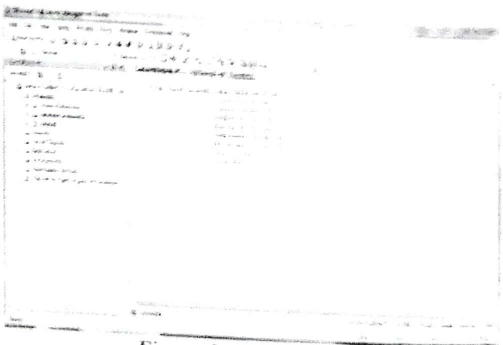


Figure 2: Creation of tables

The above figure shows the query executed to create the table in SQL Server for converting unstructured raw data to structured tabular form. The different attributes for the data are chosen keeping in mind the various categories present in the actual dataset.

Step 2: Insertion of data into the tables

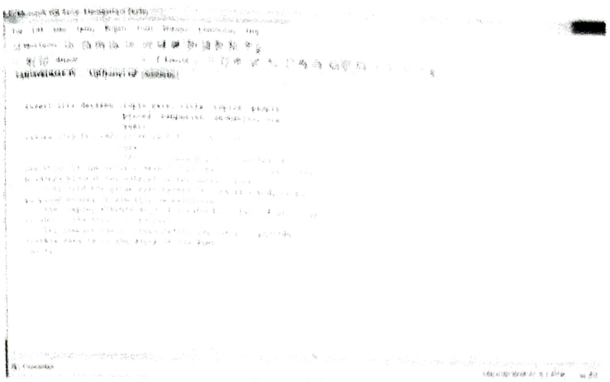


Figure 3 : Insert query

The above figure shows the query which is executed to enter the data into the tables. The various attributes are assigned vales accordingly.

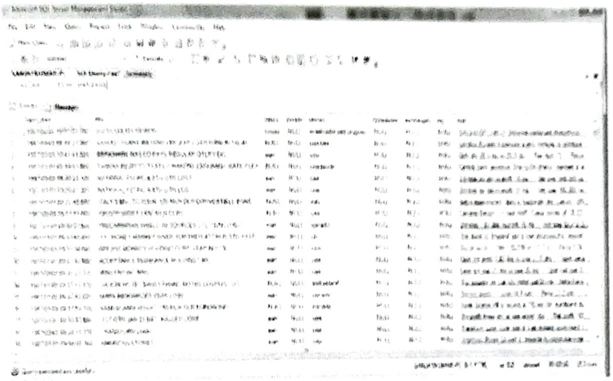
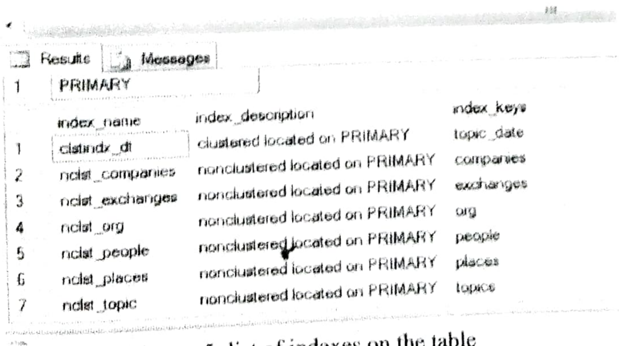


Figure 4 : Table contents

The above figure shows the content of the table. This is only a portion of the complete database. The table consists of 255 records in all pertaining to different categories.

Step 3: Creation of indexes

Following figure 2 is the list of indexes that we have created on out database:-



1	PRIMARY

	index_name	index_description	index_keys
1	clstindx_dt	clustered located on PRIMARY	topic_date
2	nclst_companies	nonclustered located on PRIMARY	companies
3	nclst_exchanges	nonclustered located on PRIMARY	exchanges
4	nclst_org	nonclustered located on PRIMARY	org
5	nclst_people	nonclustered located on PRIMARY	people
6	nclst_places	nonclustered located on PRIMARY	places
7	nclst_topic	nonclustered located on PRIMARY	topics

Figure 5: list of indexes on the table

As it can be seen from the above figure, we have made a total of 7 indexes on our dataset (table). One of them is a clustered index while all the others are non clustered index. The significance of using such indexes has already been discussed.

Step 4: Applying queries on the dataset for comparison

The various execution plans for different queries are given in a tabular form for easy understanding and comparison. The numerical values depicted in the table are in the form of CPU cycles required to perform the task. As can be noticed from table 1, the execution times in case of "clustered index seek" are lowest. The query execution is optimum in this case. Also the values for "index seek" and "table scan" are similar, this is due to the fact that in both the cases the indexes are not used.

Step 5: Converting paragraph (or document) level to sentence level.

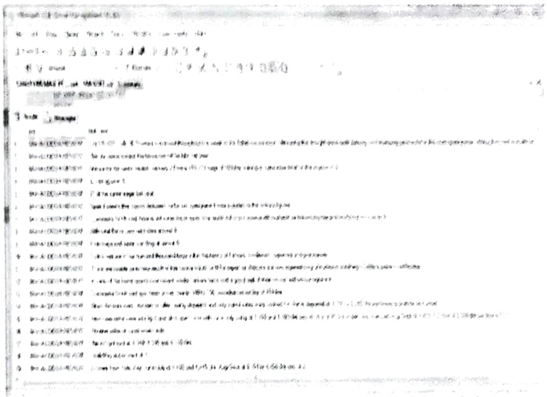
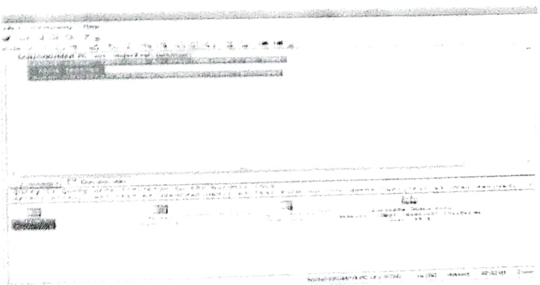


Figure 6: Sentence level fragmentation of data

The output depicted above in the figure 6 is obtained after executing the first code which is written to convert the paragraph level data to sentence level. This step is carried out in order to apply novelty mining techniques on the dataset at the sentence level rather than document level. The above output is of a temporary table which has only two attributes namely title and the text.

Step 6: Applying Novelty Mining on the dataset.

In the next step the application of novelty mining technique is



carried out. In this step the redundancy or the duplicacy in the data if present is removed by using the code developed in the form of cursors. Each sentence in the dataset is compared to the pool of existing data already present in the data and thus redundant data is ommitted and never added to this temporary table. The purpose of removing redundancy is to increase the effectiveness of search queries as the data needs not to be checked in redundant data again and again, as this consumes time and thereby degrading performance.

Step 7: Applying a search query on the dataset prior to conversion of document to sentence.

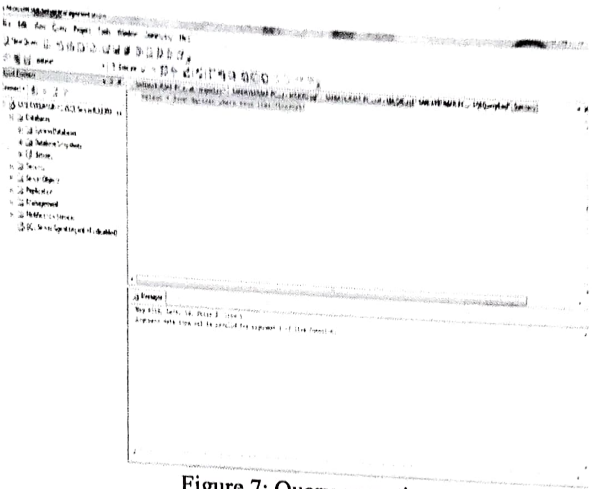


Figure 7: Query execution

As can be seen from the above figure 7 that the search of patterns is not applicable to the data directly as the data in the table is of XML form. And pattern search or text phrase search is not possible in the case of Xml attribute type. So we need to cast this data to another type such that we are able to perform search queries on the text attribute as well. The alternative solution to this problem is described in the next step.

Step 8: Casting the XML data to NVARCHAR.

The figure 8 illustrates the casting of XML data attribute to a NVARCHAR type data. The purpose of this conversion is to apply text search or pattern searches to this attribute. As depicted by the figure a search query is executed searching for the presence of phrase 'COCOA' in the dataset. And at the bottom of the figure the successful execution of the query is shown. The detailed execution times have been shown by another figure 9 shown below.

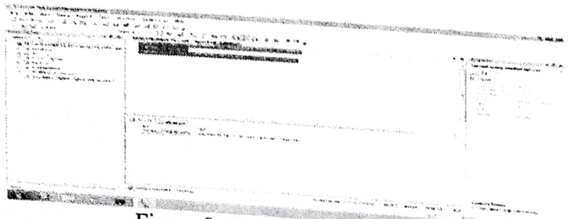
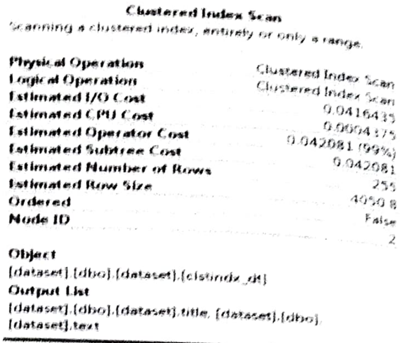


Figure 8: Query execution plan



```
              Clustered Index Scan
Scanning a clustered index, entirely or only a range.

Physical Operation                Clustered Index Scan
Logical Operation                 Clustered Index Scan
Estimated I/O Cost                           0.0416435
Estimated CPU Cost                           0.0004375
Estimated Operator Cost                0.042081 (99%)
Estimated Subtree Cost                        0.042081
Estimated Number of Rows                           255
Estimated Row Size                              4050 B
Ordered                                          False
Node ID                                              2

Object
[dataset].[dbo].[dataset].[clstindx_dt]
Output List
[dataset].[dbo].[dataset].title, [dataset].[dbo].
[dataset].text
```

Figure 9: Detailed execution plan

Step 9: Application of second code to perform search in sentence level data.

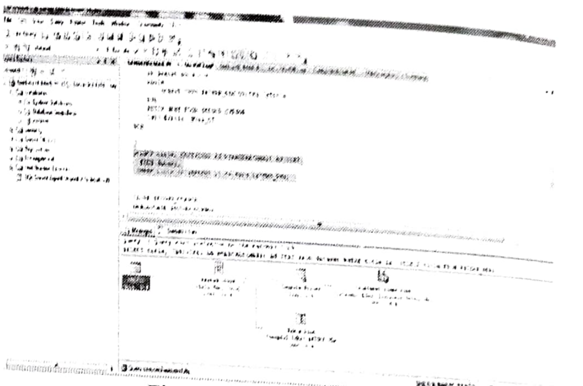
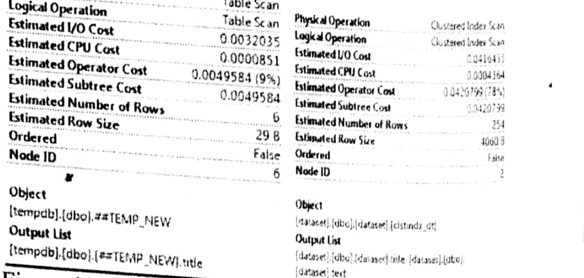


Figure 10: Query execution plan

The above figure 10 shows the application of the search in XML form data. The highlighted text in the figure is the query which is executed to display the relevant data. This method is also an alternative to the search procedure described in the previous step. The difference in this approach is that the search operation is carried out only on non redundant data; thereby overall execution time is reduced if the dataset is having redundancies present in the text. The execution plan of the query is shown above which shows the actual working of the query and transfer of control amongst the tables to display relevant data to the user.



```
              Table Scan
Scan rows from a table.

Physical Operation                      Table Scan
Logical Operation                       Table Scan
Estimated I/O Cost                       0.0032035
Estimated CPU Cost                       0.0000851
Estimated Operator Cost             0.0049584 (9%)
Estimated Subtree Cost                   0.0049584
Estimated Number of Rows                         6
Estimated Row Size                            29 B
Ordered                                      False
Node ID                                          6

Object
[tempdb].[dbo].##TEMP_NEW
Output List
[tempdb].[dbo].[##TEMP_NEW].title
```

```
              Clustered Index Scan
Scanning a clustered index, entirely or only a range.

Physical Operation                Clustered Index Scan
Logical Operation                 Clustered Index Scan
Estimated I/O Cost                           0.0416435
Estimated CPU Cost                           0.0004364
Estimated Operator Cost               0.0420799 (78%)
Estimated Subtree Cost                       0.0420799
Estimated Number of Rows                           254
Estimated Row Size                              4060 B
Ordered                                          False
Node ID                                              2

Object
[dataset].[dbo].[dataset].[clstindx_dt]
Output List
[dataset].[dbo].[dataset].title, [dataset].[dbo].
[dataset].text
```

Figure 11: Detailed execution plan

VI. Conclusion and Future Work

The proposed work uses a large dataset of news articles. An efficient way to optimize database has been proposed with indexing technique. The experimental results obtained show that the work optimizes the database with the execution time in clustered index seek and as can be noticed is lowest out of all other attributes. Also the values of Index seek and Table scan are similar, as both do not consider indexes. With the proposed work the effectiveness of optimization has been studied experimentally. Further investigation to the topic reveals that novelty mining with database optimization can give good results.

The results obtained from the experiments conducted show that the execution time in case of sentence level search as well as cast search is similar. This is due to the fact that the dataset does not contain redundancies Moreover the Document to sentence conversion is also successfully carried out with the help of the proposed algorithm. Thus the proposed optimization and novelty mining algorithms are efficient. Mining of documents for novel information is successfully accomplished by removing redundancy or duplicity from the data.

VII. Acknowledgements

We take this opportunity to express our sincere thanks and deep gratitude to all those who extended their wholehearted co-operation and have helped us in completing this work successfully. We express our sincere thanks to Mr. Suyash Gupta, DBA (HCL Technologies) for his encouragement and valued suggestions.

VIII. References

[1] A.T. Kwee, and F. S. Tsai, "Database Optimization for Novelty Mining of business blogs", Elsevier Expert Systems with Applications vol. 38, pp. 11040-11047, 2011

[2] J. Allan, C. Wade, and A. Bolivar, "Retrieval and Novelty Detection at the Sentence Level", SIGIR'03, ACM 1-58113-646-3/03/0007., Toronto, Canada, July 28–August 1, 2003

[3] F. S. Tsai and K. L. Chan, "Detecting Cyber Security Threats in Weblogs Using Probabilistic Models", Springer – Verlag Berlin Heidelberg , LNCS 4430, pp. 46-57, 2007.

[4] H. Cui ,M.Y. Kan and T.S. Chua, "Unsupervised Learning Of Soft Patterns For Generating Definitions From Online News", ACM 1-58113-844-X/04/0005, WWW 2004, May 17–22, 2004.

[5] F. S. Tsai, Y. Zhang, "D2S: Document-to-sentence framework for novelty detection", Received: 18 June 2009 / Revised: 22 July 2010 / Accepted: 11 December 2010© Springer-Verlag London Limited 2010

[6] S. Deerwester, S. T. Dumais, R. Harshman, "Indexing by latent semantic analysis", Journal of the American Society for Information Science, vol. 41,pp. 391, 1990

[7] F. S. Tsai and K. L. Chan (2010), "Redundancy and novelty mining in the business blogosphere", The Learning Organization, vol. 17 ,Iss: 6, pp. 490 – 499, Emerald Article.

[8] R. Feldman, The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data, Israel James Sanger ABS Ventures, Waltham, Massachusetts.

[9] The dataset source: http://kdd.ics.uci.edu/databases/reuters21578

[10] Porter-Stemmer algorithm for preprocessing of dataset.

[11] http://msdn.microsoft.com/en-us/library/aa964133(v=SQL.90).aspx

[12] http://en.m.wikipedia.org/wiki/Database_index