Model for Automatic Textual Data Clustering in Relational Databases Schema

Wael M.S. Yafooz , Siti Z.Z. Abidin , Nasiroh Omar *and* Rosenah A. Halim

Faculty of Computer and Mathematical Sciences, UiTM Shah Alam, Selagor, Malaysia

Waelmohammed1@hotmail.com,{zaleha,nasiroh,rosenah}
@tmsk.uitm.edu.my

Abstract. In the last two decades, unstructured information has become a major challenge in information management. Such challenge is caused by the massive and increasing amount of information resulting from the conversion of almost all daily tasks into digital format. Tools and applications are necessary in organizing unstructured information, which can be found in structured data, such as in relational database management systems (RDBMS). RDBMS has robust and powerful structures for managing, organizing, and retrieving data. However, structured data still contains unstructured information. In this paper, the methods used for managing unstructured data in RDBMS are investigated. In addition, an incremental and dynamic repository for managing unstructured data in relational databases are introduced. The proposed technique organizes unstructured information through linkages among textual data based on semantics. Furthermore, it provides users with a good picture of the unstructured information. The proposed technique can rapidly and easily obtain useful data, and thus, it can be applied in numerous domains, particularly those who deal with textual data, such as news articles.

Keywords- relational databases, unstructured data, document clustering, query efficiency, textual data

1 Introduction

Unstructured information presents significant challenges in information management. Given their unorganized form, rapid management and retrieval of such information, which is essential in providing users with knowledge, is difficult[1, 2]. In addition, no rule or constraint exists in handling unstructured information. The amount of unstructured information increases as a result of the high reliance of users on digital data in almost all forms of daily tasks because this format is more secure, less storage space, and easy to retrieve as compared to hard copies[3]. Unstructured data can be found in relational databases such as news articles, personal data and textual documents. In spite of, Relational Database Management Systems (RDBMS) are powerful and robust data structures used in managing, organizing, and retrieving data [4, 5]. However, such systems contain massive amount of unstructured data, which are mea-

T. Herawan et al. (eds.), *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Lecture Notes in Electrical Engineering 285, DOI: 10.1007/978-981-4585-18-7_4,

ningless and difficult to deal without proper organization if left unorganized. As a result, retrieval of significant information or pertinent knowledge becomes a challenge.

Few attempts have been made to deal with unstructured data in RDBMS. These attempts focus only on named entity and on extracting structured information often hidden in unstructured data. Such methods managed unstructured data in the database schema itself as an incremental repository, in databases structure (schema) [5-7], to answer a structured query [8-11] or retrieved such data by keyword search [11-15]. Such methods fail to represent the entire collection of textual documents (corpus) in meaningful clusters, which can help users acquire knowledge regarding the corpus. These methods also do not consider the semantic relation among unstructured data stored in relational databases. Moreover, the aforementioned methods require extra scripting and programming to manage and represent unstructured data in appropriate formats. Thus, these methods are time consuming and labor intensive.

This paper is an extension of our previous work [16]. The most common methods used for managing unstructured data in relational databases are investigated. In addition, an incremental and dynamic model for managing and clustering unstructured (textual) data is introduced. This model automatically processes data when the user loads textual documents into a relational database. The proposed technique is performing automatic textual data clustering and linking. By applying this concept, frequent term, which is used in document clustering, and named entity, which is used in information extraction, are presented. In addition, the semantic of the words in clustering process is included using WordNet database[17]. In this manner, the user can obtain the knowledge and useful data clusters based on the semantic relation among unstructured data. In addition, the efficiency of query processing is improved when retrieved the clustered data. Furthermore, the user is not required to develop extra programming for executing textual data the clustering on desktop application due to clustering process is performed automatically in database schema.

The rest of this paper is organized as follows. Section 2 presents related studies. Section 3 describes the proposed technique, that is, an incremental and dynamic repository for managing unstructured data in relational databases. Section 4 provides the conclusion.

2 Related studies

Relational databases contain massive amounts of unstructured information. Few attempts have been made to handle such information using information extraction techniques [18]. This challenge is addressed using the canonical link among named entities often found in articles because such entities exist in different formats or varieties in database records proposed by [7]. The proposed technique is executed in two processes. First, the named entities are recognized in the articles. Second, matching is performed by introducing the canonical link as the foreign key, which matches database records.

In [5], another method for extracting and storing structured information into a table is introduced. The table can be used for keyword search. In addition, three operators, namely, extract, cluster, and integrate, are presented. These operators can be used by the database administrator to manage extracted information. All the aforementioned methods focus on dealing with unstructured data in the same database schema. However, a method for storing extracted information is introduced in [6] by developing an intermediate database, called parse tree database (PTDB), and a query language, called parse tree query language (PTQL).

PTDB stores immediate data from the extracted information and works as intermediate between user and relational database. This process is the initial step in the proposed technique. PTQL is the query language used to retrieve extracted information from PTDB. This language decomposes retrieved queries into keyword-based queries and Structured Query Language (SQL). Recently, statistical technique based classification [19] and integration time series analytical data based on forecasting [20] are introduced for managing such data. Majority of aforementioned approaches deal with unstructured information in a low-level database schema. Another technique for dealing with such information is using query as a top-level database. This method is called answering the structured query, and the examples include SCORE[8, 9] , EXDB[21, 22] , Avatar [11],and SCOUT [10, 23]. Keyword search technique [12-15] is another method to manage unstructured data.

In a commercial database, such as Oracle®, the introduced oracle text [24, 25] which is consists of two main types of classification and clustering. Classification is based on a predefined class. Meanwhile, k-means and its variants clustering algorithms are used for clustering [26]. The numbers of clusters are needed to parameterize. In addition, text index on all textual data is required to create. This method is time consuming and cannot produce good quality data clusters. In addition, this is required users to enter the number of clusters in order to perform the clustering. Thus, user' should have a prior knowledge is on the textual data collection. Furthermore, several tedious steps are required before the clustering process. Table 1 summarizes the research works that focus on managing unstructured data in RDBMS in database schema or in query level.

Table 1. summarize of unstructured data management in RDBMS

Approach	Style	Strategy	Research work
Database	Inside Schema	Named Entity	[7]
Schema	Intermediate	Named Entity	[6]
	RDBMS		
	Inside Schema	Structured Information	[5]
	Inside Schema	All words	[24, 25]
Query Based	Decompose		[11-15]
	SQL	Common Words	

Most of aforementioned methods are based on extracting the structured from unstructured information by using named entity techniques in databases schema (Low level). Such methods are used only for building information extraction architecture within relational databases, while query based (high level) methods are running on top of the relational database. Such methods do not concern about organizing the actual data. These methods focus on retrieving textual documents from text databases. The keyword search techniques simplify the process of structured query language over relational databases only. However, they are time consuming and need extra tools or expert user to perform further processing. In addition, such methods do not concern in managing unstructured data from the start of the storing process.

3 Incremental and Dynamic Repository and Semantic Query

In this section, the proposed technique for managing unstructured information in relational databases is presented. This technique includes an incremental and dynamic repository and semantic query.

3.1 Incremental and Dynamic Repository

Incremental and dynamic repository is a method for organizing and clustering textual data. This technique is embedded within the database schema. The technique proposed in this paper is divided into three stages, namely, data loading, data filtering, and data clustering.

Data loading.

The first stage is data loading. During this stage, the user enters textual data into a database table. Before the data is stored, several steps are performed to rearrange the textual data. These steps include stemming that converts words into their source, stop words remove and noise cleaning, such as HTML or XML tags. In this step, there are two versions are original and cleaning of textual data. In original version, textual data which is stored in a specific data record while cleaning version is hold the textual document after rearrangement process. This version used to extract and stored such information in the incremental repository (database table) to be continuously used further processing. When all processes are completed, information filtering begins.

Data filtering.

During this stage, concepts from information extraction and textual document clustering [27, 28] are applied. Information extraction converts textual data into structured information by extracting the named entity. Furthermore, in certain cases when named entity is ambiguous, tools such as IRC-names [29] and gazetteers are applied. Textual document clustering adopts the clustering approach in which common words are

based on textual documents. Extracted information is stored in the incremental repository for usage in data clustering.

Data clustering.

Clustering is automatically executed based on common words and the named entity [30, 31]. In this manner, the process is faster when compared to common techniques used in traditional textual clustering methods. In addition, the clustering process is dynamic and its relies on some parameters entered by user at the first time such as minimum support of words.

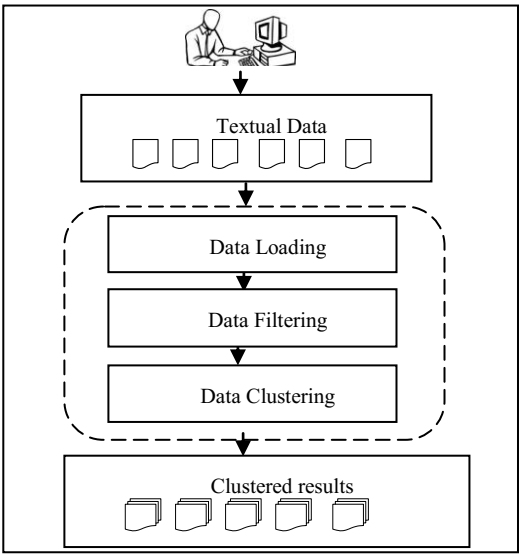
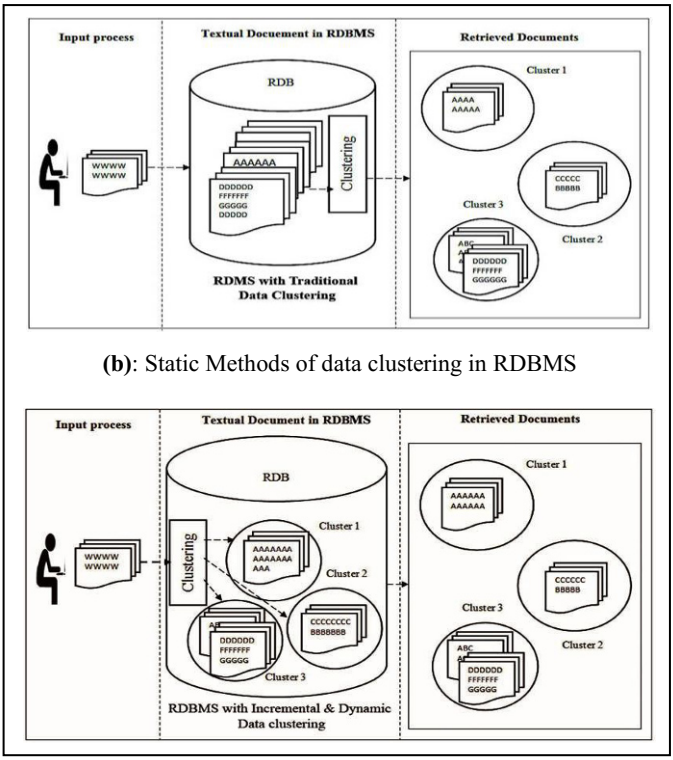


Fig. 1. Stages of incremental and dynamic repository

Minimum support of words used for disjoint data clusters to prevent the overlapping between them due to the representation of first cluster in hierarchical view. Hierarchical representation for textual document is preferable compare to partitonal representation[3]. Thus, textual document can be viewed in the form of topic and subtopic structure. Therefore, the user can obtain useful information when retrieving the textual document into meaningful clusters. Such data clusters can be used in many text domains such as text summarization, topic detecting and tracking, personal information management and managing extracted information.

Fig.1 demonstrates the three main stages of incremental and dynamic repository. The data clustering is performed automatically and incrementally in the database schema of RDBMS. Thus, the user does not need to another application for performing data clustering. In contrast, the traditional methods of clustering that execute in batch mode that gather all textual data before perform the clustering process. However, such methods are labor intensive and time consuming. Fig. 2 (a and b) show the comparison between traditional methods and incremental and dynamic technique.



(b): Static Methods of data clustering in RDBMS

(b): Incremental and dynamic data clustering in RDBMS

Fig. 2. Clustering Process

Fig.2 demonstrates the method of data clustering in RDBSMS. Both methods consist of three stages. First, the user enters textual documents to RDBMS in input process. In the second stage, in static method the textual data is directly stored into database in the dynamic method, the data clustering is performed automatically based on the extracted information as discussed in section 3.1. In third stage, the traditional method is performed after whole texts are collected. However, in the proposed techniques the data cluster retrieved automatically without the need to do the clustering process. Thus, the time consumed is reduced. In addition, the searching process can be minimized by creating text index of the extracted keywords.

The primary benefits of the proposed technique are as follows. (1) The relationship among unstructured data within massive relational databases is determined. The discovered relationship is not only based on separate words, but also on semantic relation, which can be extracted from the WordNet database. (2) The user can obtain useful information and knowledge from the extracted information representing the textual documents. These extracted information known as the shortcut for textual documents. (3) Retrieval precision is improved because data are already clustered based on meaning. Furthermore, the proposed technique does not need user intervention except only at the first time of implementing the software package.

The software package is simple to be execute by a normal user and it can be included by database administer at the design or later. The proposed techniques provide structured query language (SQL) operator. The SQL operator can be used to retrieve the data clusters based on the semantic meaning of words as will be discussed in next section.

3.2 Semantic Operator in SQL

The semantic operator, which can be used in SQL query, is introduced to increase precision and recall. These factors are important in achieving good quality results. By using the semantic operator, the required word in a user query is obtained. The synonyms of the word are searched in the WordNet database[17]. In this manner, a list of words (required word + its synonyms) used to retrieve textual data cluster are represented by keywords are already extracted and stored in the incremental repository. These keywords act as the shortcut to retrieve specific textual document for example Table 2, presents table named “Articles,” in database that contains four columns, namely “id,” “articles,” , “extracted_info” and "data_cluster". The “id” column holds the identifier name of the articles or their serial number. The “articles” column contains actual textual data. The “extracted_info” column contains data extracted from the original file. The “data_cluster” column holds cluster identifier. For example, a user needs to select data from the “Articles” table by issuing a query on extracted information under the word “perform.” However, the user uses a different keyword, that is, “execute” (typed in normal case) instead of “perform,” therefore, the retrieved result is null. In the proposed technique, the semantic operator will obtain the synonyms of “perform” from the WordNet database, and then, it will search for these synonyms. In this case, the results will show existing database records and the textual document will be retrieved. The SQL format is as follows:

*SELECT * FROM articles where SEMANTIC ('perform');*

Table 2. Example of Semantic Query (Articles database table)

Id	Article	Extracted_Info	Data_Cluster
1	Microsoft announces about... Bill Gates.... in conference .. USA... perform	Microsoft, USA, Bill Gates, perform	C1
2	Oracle introduce.... IBM.....in united kingdom.... introduce	Oracle, united kingdom, IBM, introduce,	C2
3	Bill Gates said that..... evaluate	Bill Gates, evaluate, achieve	C1

The SEMANTIC operator can be used in the procedural structured query language which is provided in any relational databases. By using such language function or procedure can develop with passing parameter from the user. The user parameter is actual words that need to find its relevant in the relational database .The parameter can be presented not only in one word but it can be in many forms of words. In addition, the operator can be used to perform semantic integration between the unstructured information and keywords from user or applications. In this manner, the user can obtain a full picture about the content of textual documents. Fig. 3 shows the semantic query processes which begin with the user query with include the "SEMANTIC" operator in SQL command.

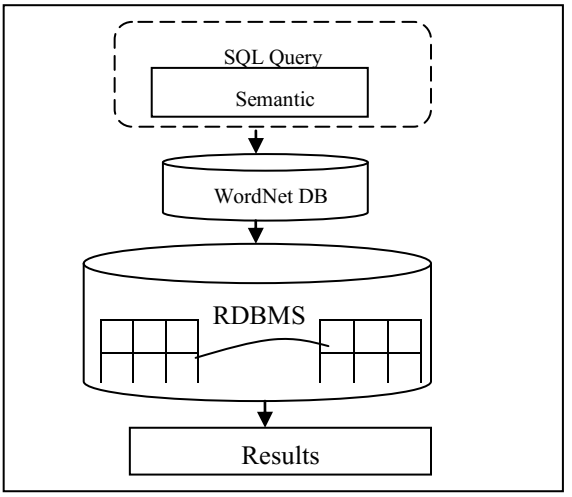


Fig. 3. Semantic Query Structure

The words are entered in semantic operator and sent to WordNet in order to obtain its synonyms. The synonyms used to look for the words that stored in the extracted columns. Thus, the query retrieves the required data in data clusters or only the ordinary requested data.

4 Conclusion

In this paper, an incremental and dynamic repository for managing unstructured data in relational databases is introduced. The system architecture of the proposed technique is described. This repository is created by using information extraction and textual document clustering techniques. Information extraction is conducted incrementally, whereas clustering is conducted dynamically. This method of retrieving data clusters is extremely fast as compared to traditional methods, in which an entire collection of textual documents (corpus) is clustered at one time. Such methods are time consuming. Furthermore, traditional methods are performed on top of a database using other applications with extra scripting and programming. Each time the user

requests to see a certain data cluster, the entire clustering process needs to be repeated by extract the same collection of textual documents. In contrast, the proposed technique is embedded within the schema of relational databases. This technique presents the relationship among textual documents based on semantics. Moreover, it improves query efficiency when retrieving data clusters of textual documents. It proposed is useful for text domain researchers and developers, such as those who involve in online news services.

Acknowledgment

The authors wish to thank Universiti Teknologi MARA (UiTM) for the financial support. This work was supported in part by a grant number 600-RMI-/DANA 5/3/RIF (498/2012).

References

1. Doan, A., et al., Information extraction challenges in managing unstructured data. SIGMOD Record, 2008. Vol. 37, No. 4.
2. Doan, A., et al., The case for a structured approach to managing unstructured data. arXiv preprint arXiv:0909.1783, 2009.
3. Li, Y., S.M. Chung, and J.D. Holt, Text document clustering based on frequent word meaning sequences. Data & Knowledge Engineering, 2008. 64.1: p. 381-404.
4. Blumberg, R. and S. Atre, The problem with unstructured data. DM REVIEW, 2003. 13: p. 42-49.
5. Chu, E., et al., A relational approach to incrementally extracting and querying structure in unstructured data. Proceedings of the 33rd international conference on Very large databases, 2007. VLDB Endowment.
6. Tari, L., et al., Parse Tree Database for Information Extraction. IEEE TRANSACTIONS ON KNOWLEDGE and DATA ENGINEERING, 2010.
7. Mansuri, I.R. and Sarawagi, Integrating unstructured data into relational databases. Data Engineering, ICDE'06. Proceedings of the 22nd International Conference on. IEEE, 2006.
8. Roy, P., et al., Towards Automatic Association of Relevant Unstructured Content with Structured Query Results. Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, 2005.
9. Roy, P. and M. Mohania, SCORE: symbiotic context oriented information retrieval. Advances in Data and Web Management. Springer Berlin Heidelberg, 2007: p. 30-38.
10. Jain, A., A. Doan, and L. Gravano, Optimizing SQL Queries over Text Databases. Data Engineering, . ICDE . IEEE 24th International Conference on. IEEE, 2008.
11. Kandogan, E., et al., Avatar Semantic Search: A Database Approach to Information Retrieval. SIGMOD , Chicago, Illinois,USA, 2006: p. 790-792.
12. Agrawal, S., S. Chaudhuri, and G. Das, DBXplorer: A System for Keyword-Based Search over Relational Databases. Data Engineering. Proceedings. 18th International Conference on. IEEE, 2002.
13. Hristidis, V. and Y. Papakonstantinou, Discover: Keyword search in relational databases. Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002.
14. Li, G., et al., EASE: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-structured and Structured Data. Proceedings of the ACM SIGMOD international conference on Management of data, 2008.

15. Luo, Y., W. Wang, and X. Lin, SPARK: A Keyword Search Engine on Relational Databases. Data Engineering. ICDE. IEEE 24th International Conference on. IEEE, 2008.
16. YafoozA, W.M.S., S.Z. Abidin, and N. Omar, Towards automatic column-based data object clustering for multilingual databases. Control System, Computing and Engineering (ICCSCE), IEEE International Conference on. IEEE, 2011.
17. Miller, G., WordNet: A Lexical Database for English. Communications of the ACM 1995. 38.11: p. 39-41.
18. Sarawagi, S., Information Extraction. Foundations and Trends in Databases, 2008. Vol. 1, No. 3 (2007): p. 261–377.
19. Koc, M.L. and C. R´e, Incrementally Maintaining Classification using an RDBMS. Proceedings of the VLDB Endowment, 2011. Vol. 4, No. 5.
20. Fischer, U., et al., Towards Integrated Data Analytics: Time Series Forecasting in DBMS. Datenbank Spektrum 2013. 13.
21. Cafarella, M.J., et al., Structured querying of Web text. 3rd Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, California, USA, 2007.
22. Cafarella, M.J., Extracting and Querying a Comprehensive Web Database. Proc. of the 4 th Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA., 2009.
23. Jain, A., A. Doan, and L. Gravano, SQL Queries Over Unstructured Text Databases. Data Engineering. ICDE, IEEE 23rd International Conference on. IEEE, 2007.
24. Text, O., 11g Oracle Text Technical White Paper. 2007.
25. Text, O., an oracle technical white paper. 2005.
26. Jain, A.K., N. Murty, and P.J. Flynn, Data Clustering: A Review. ACM computing surveys (CSUR), 1999. 31.3: p. 264-323.
27. Su, C., et al., Text Clustering Approach Based on Maximal Frequent Term Sets. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA, 2009.
28. Vishal Gupta, G.S.L., A Survey of Text Mining Techniques and Applications. JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, 2009. VOL. 1, NO. 1.
29. Steinberger, R., et al., RC-NAMES: A Freely Available, Highly Multilingual Named Entity Resource. In RANLP 2011: p. pp. 104-110.
30. YafoozB, W.M.S., S.Z. Abidin, and N. Omar, Challenges and issues on online news management. Control System, Computing and Engineering (ICCSCE),IEEE International Conference on., 2011.
31. Fung, B.C.M., K. Wangy, and M. Ester, Hierarchical Document Clustering Using Frequent Itemsets. Proceedings of the SIAM international conference on data mining, 2003. 30. No. 5.