Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

OPEN ACCESS Research Journal of Information Technology

ISSN 1815-7432
DOI: 10.3923/rjit.2017.7.17

Research Article
Model of Textual Data Linking and Clustering in Relational Databases

[1,2]Wael M.S. Yafooz

[1]Department of Computer Science, Faculty of Computer and Information Technology, Al-Madinah International University (MEDIU), 40100 Shah Alam, Selangor, Malaysia
[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Shah Alam, Selangor, Malaysia

Abstract

Background: A huge reliance on computer usage in everyday life leads to the continuous increase of large data applications in the form of textual data. The data are reposited to produce meaningful information. Therefore, databases become a backbone in most application software for organizing data into structured form. The structured information provides users with comprehensible knowledge. However, dealing with a large amount of textual data leads to two basic issues; insufficient query processing performance and inaccurate information retrieval. Attempts have been made to resolve both issues by database clustering techniques and textual document clustering. Nevertheless, most of the attempts require several stages of tedious programming scripts in constructing software applications that are external to databases. **Materials and Methods:** Therefore, this study proposes a Textual Virtual Schema Model (TVSM) to structure extracted textual data, while performing automatic column based information clustering in the internal structure of a relational database. Furthermore, a similarity measurement method is introduced to obtain high accuracy data clusters. An experiment has been conducted on textual Reuters's corpus, WAP and classic dataset. Then, the clustering results are validated by measuring F-measure, entropy and purity. **Results:** The results show linkages between structured textual data and unstructured information, high performance of query processing and time improvement in document clustering with accurate clusters. **Conclusion:** This model envisages a beneficial and useful approach for various domains that involves a large amount of textual data such as document clustering, topic detecting and tracking, document summarization, personal data management and information retrieval.

Key words: Unstructured data, relational databases, textual document clustering, information extraction, information management, database clustering

Received: April 03, 2014 Accepted: November 18, 2016 Published: December 15, 2016

Citation: Wael M.S. Yafooz, 2017. Model of textual data linking and clustering in relational databases. Res. J. Inform. Technol., 9: 7-17.

Corresponding Author: Wael M.S. Yafooz, Department of Computer Science, Faculty of Computer and Information Technology, Al-Madinah International University (MEDIU), 40100 Shah Alam, Selangor, Malaysia

Competing Interest: The author has declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

A huge reliance on computer applications in everyday life leads to the continuous increase of large textual data usage. Commonly, textual data will increase in size at a tremendous rate[1]. The data is a source of knowledge transfer that can be found in online media such as news articles, discussion forums and personal pages. Such data normally contains unstructured information which is vital and necessary for many applications[2]. However, it is hard to discover, detect and extract the useful information.

Information extraction techniques play a significant role in performing the extraction of structured information (data). Many approaches have been introduced which can be categorized into a knowledge engineering/rule-based system and training-based[3-5]. In most cases, the extracted information are reposited in an organized form in relational databases[6,7].

There are many issues in managing unstructured textual data in relational database[8,4]. The most critical issue is converting such huge unstructured data[8] into structured[1]. There are several trends in managing unstructured data in a database that include developing new data model[9-11] within the database and query based techniques in database applications. Query based techniques can be in the form of "Extract-Then-Query"[12,13]"Query-Then-Extract" or "Keyword Search[14,15]". Recently, another trend has emerged for a new generation of databases called NoSQL (Not only SQL known as non-relational databases). These databases have been concerned with managing unstructured data in distributed environments. The examples of such databases include Cassandra and DBmond.

Relational database (RDB) stores structured information and provides significant knowledge to users. It is the best repository for data retrieval, deletion and modification[8]. Textual data (structured or unstructured) are stored in a systematic manner based on meta-data. However, the increase in textual data raises two problems; insufficiency query processing performance and inaccurate query results. First, insufficiency query performance has gained more attention in reducing access of a secondary storage (hard disk) for data retrieval. Second, inaccurate results in query due to a massive unstructured information within the textual data. Query performance can be enhanced by database partitioning techniques that concern on meta-data (attributes and records)[16].

Currently, there are two methods in order to retrieve the required structured information of actual data in the database. The first method is via an extra application (external to the database) that performs a database query to retrieve the stored unstructured information. After gathering all the required information, document clustering techniques[17] are carried out to cluster and convert such information into a structured form for later retrieval. The second method is by having a clustering application (internal to the database) that gathers all the information and performs document clustering in a batch mode for database query. Both methods are time consuming and perform same clustering process[18,19] repeatedly[6].

The common clustering techniques for document clustering are term frequent[20-28], concept-based[22,29-31], named entity-based[32-34] and classical document clustering[35]. Clustering is desirable in document organization, especially for dynamic information environments such as the world wide web or stream of newspaper articles[36,37].

In this study, we introduce a novel technique, namely a Textual Virtual Schema Model (TVSM) for handling textual data in a relational databases extended[38,39]. Any textual data will be automatically extracted from online information sources and at the same time, automatic document clustering is performed on the actual data within the database structure. Therefore, this study contributes in three folds. Firstly, extracted unstructured data is transformed into structured data. Secondly, the actual data in the database are clustered for faster text mining. Thirdly, a novel similarity measure is introduced in the document clustering for query accuracy. As a case study, an experiment is conducted on Reuters data set with more than one thousand documents. This study is one of its kind which able to automatically extract and dynamically cluster any information before storing such information in the database. It is part of the internal structure of a relational database management system. Thus, TVSM produces faster query performance with better data accuracy as compared to the previous approaches.

MATERIALS AND METHODS

This study proposes a novel approach for managing relationship between actual data of unstructured information in RDBMS. It is introduced as a layer to be added to the database schema design in order to organize textual data called a Textual Virtual Scheme Model (TVSM). It can perform automatic semantic textual data linking and clustering on a storage medium for relational databases. In addition, TVSM discovers hidden semantic relation between textual documents. It can be developed as a package to be used in any database scheme. Furthermore, TVSM provides quick extraction, data arrangement and data clustering based on pattern similarities. The clustering process on the textual

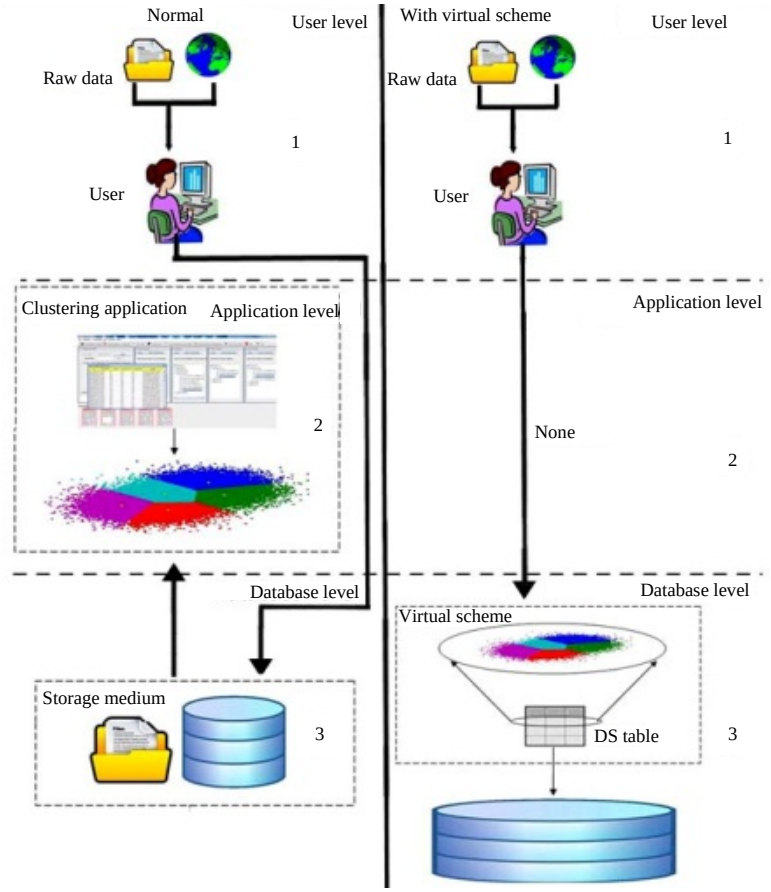

Fig. 1: TVSM and traditional methods

documents is executed by using a proposed similarity measurement. Additionally, it achieves high quality data clusters and improves the efficiency of query processing through back end query clustering. Moreover, TVSM converts unstructured information into a structured data form. Figure 1 illustrated the difference between a normal logical model and TVSM.

The TVSM model consists of three levels, user, application and database. At the user level, there is no difference between the two because the raw data from a user is always necessary. However, at the application level, the clustering application is not needed in the TVSM model due to its automatic clustering when any record is created. In the database level, the normal model only store and provide data for extraction while the TVSM performs all clustering activities based on name entities and frequent terms for future data extraction.There are some commercial databases perform internal data clustering such as Oracle and Sybase, which can referred to as traditional textual data clustering. Figure 2a and b illustrated the difference in clustering between traditional textual data and TVSM methods in relational databases.

The traditional textual clustering works as for batch mode, where it performs clustering iterations repeatedly on the same dataset. In addition, it needs a lot of coding and tedious steps to perform the clustering process. Furthermore, traditional methods suffer from the high dimensionality of data due to referring to all terms exist in the textual document. Additionally, the number of clusters should be set as a predefine parameter prior to the clustering process. In contrast, TVSM performs textual data clustering. In addition, it does not need any predefine parameters from users. Furthermore, TVSM uses Named entity and most term frequent with minimum support of words[40]. User can execute SQL query to display content of the cluster with the percentage of similarity between textual documents or with cluster description. Cluster description is a set of named entities and term frequents that represent all textual documents. Often, the last step of clustering is cluster representation; the cluster representation can be partitional or hierarchical shape by using some simple SQL commands.

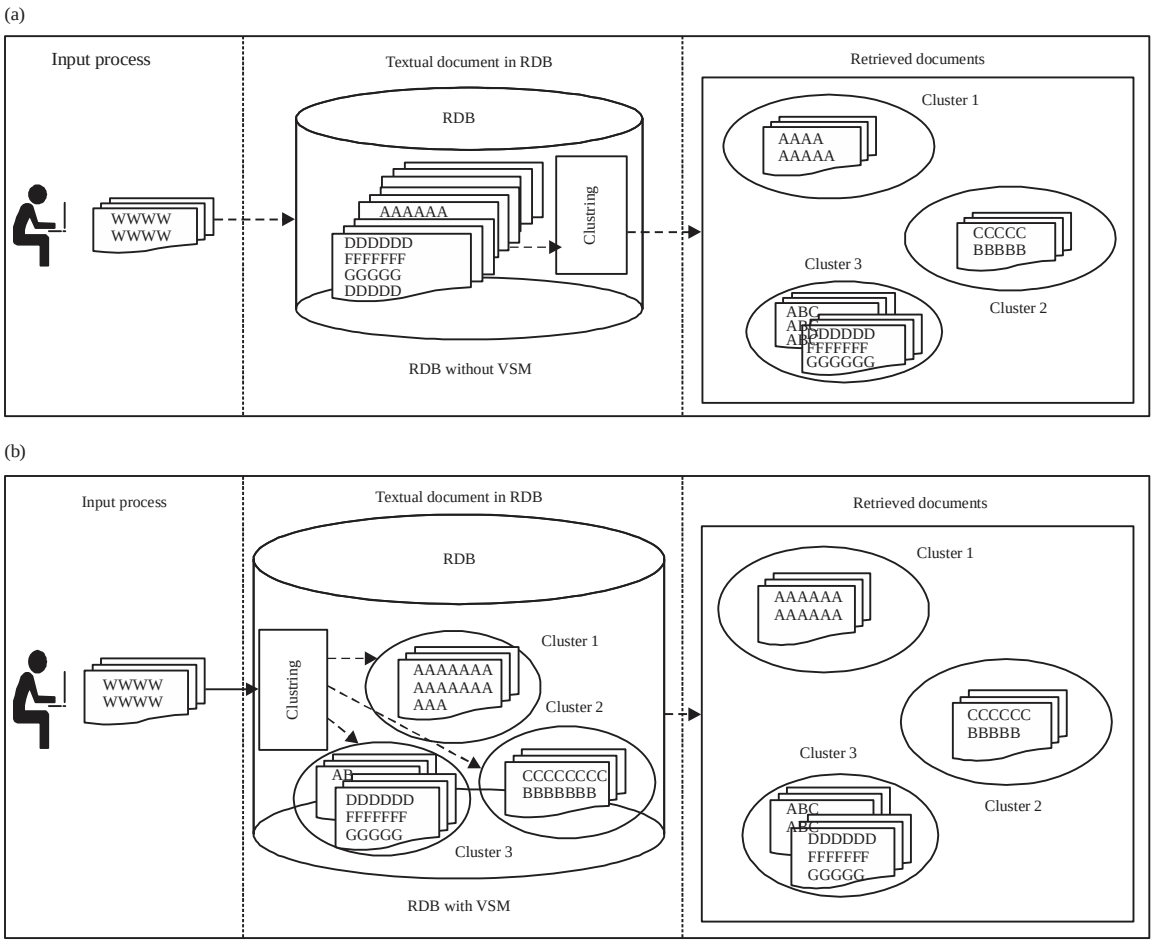

Fig. 2(a-b): Clustering methods with/without TVSM, (a) Clustering in traditional methods and (b) Clustering with TVSM

System architecture: The TVSM consists of two main components which are Data Acquisition (DA) as the first phase and its second phase is Textual Data Management (TDM) as shown in Fig. 3. In DA, ordinary users enter any format of textual data (structured or unstructured) to database table. The storing process involves several steps in TDM. In TDM, there are two main steps, term mining and data clustering. Term mining performs document pre-processing, named entity extraction, frequent term search and semantic linkage. Next, data clustering clusters the textual input data by executing two processes, similarity measure and cluster description.

Document pre-processing is a process of cleaning and rearranging the textual data. In document cleaning filtration, stop words removal, stemming and document representation is carried out. Filtration removes any format or noise from textual document such as HTML tags or XML. Stop words removal removes the list of stopwords such as 'a', an',' the' according to standard list[41]. Stemming converts words to source by using porter algorithm[42]. Document representation uses vector space model, which used for represent document in form of words and its frequencies with added columns for named entity and frequent term. The output of this process will be used in named entity extraction process.

Named entity extraction is a process to extract the named entity such as person, organization and place, based on NER-Stanford[43,44]. The extracted named entity is stored along with its frequencies. Next, frequent term search process mines and selects the frequent terms from textual data according to its frequencies known as minimum support. All selected frequent terms are used in next stage to get the semantic.

Semantic phase deals with the synonym for selected frequent terms from word net database[45] that enables TVSM to discover the hidden semantic relation between textual data (documents). Thus, the synonym of selected frequent terms and named entity are used as input for matching process to find an existing textual data cluster. If the cluster found, it is considered as the selected cluster for that textual document. Otherwise, a new textual data cluster is created. The matching process is based on the similarity measure which will be

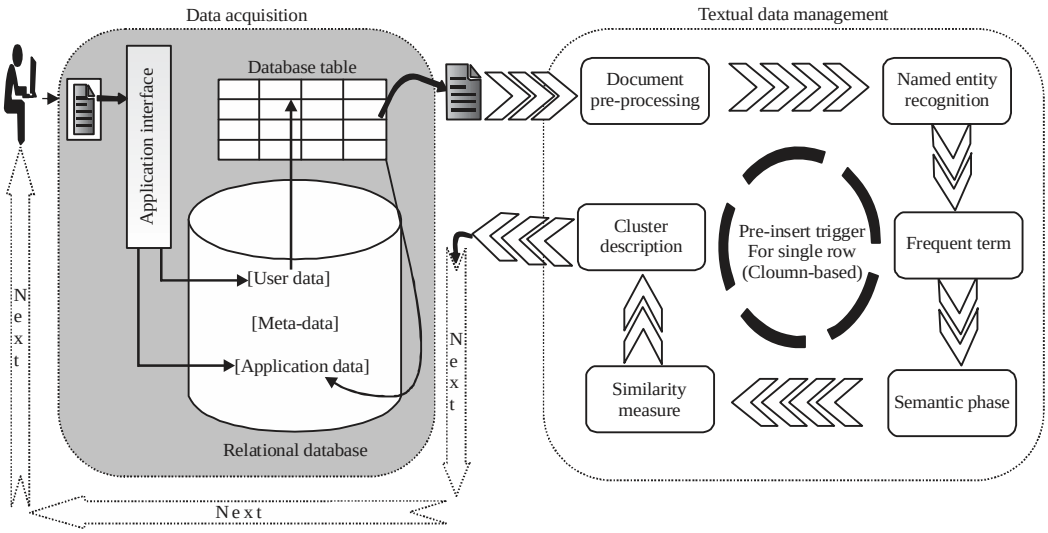

Fig. 3: TVSM system architecture

discussed in this study. The last process is adding the selected frequent terms and named entities to the cluster description. Cluster description is the collection of selected frequent terms and named entities that describe the textual data cluster. It is used in the process of similarity measure for incoming text data.

Mathematical framework: Clustering textual documents requires similarity measure for producing a highly accurate results[46,47]. Traditionally, the similarity measure is usually performed by statistical methods that based on the VSM, such as cosine similarity[48], Euclidean distance and Jaccard coefcient[45]. In addition, the similarity can also be measured by frequent item/term set in document[25,27]. However, term frequent suffers from a high dimensionality of data that is later reduced by maximal frequent terms[25,49]. All the aforementioned similarity measures concentrate to solve a high dimensionality problem but the "goodness" of data clusters quality is not considered. Therefore, Malik *et al.*[50] and Shehata *et al.*[51] focus on producing a high quality of textual data clusters. Malik *et al.*[50] use a closed interesting terms while Shehata *et al.*[51] use concept-based approach. Another similarity measures focus only on named enitiy[32-34,52]. Montalvo *et al.*[52] use a named entity to cluster bilingual textual documents and their results show that named entity outperforms the results of traditional methods. Meanwhile, Cao *et al.*[34] concentrate on the ambiguity of the named entity, specifically on geographical information to show that the mentioned name belongs to the appropriate entities. Unfortunately, relying only on the named entity as the similarity measure is not sufficient enough to produce a high accuracy of textual document clusters[53]. Therefore, we propose a new similarity measure based on all available named entities along with the frequent terms in textual document (data). Our approach ensures that the quality of textual document clusters can be obtained.

In the clustering process, there a collection of documents, denoted by D that needs to be clustered into one or more clusters represented by C. The documents in a cluster should similar properties. Thus, clustering process depends on similarity measurement between pair documents or document clusters. Our similar measurement can be represented by the following terms.

Definition 1: There exists a document cluster (C) and a collection of document (D_{all}) where, D_{all} can be clustered into one or several C. Therefore, $D_{all} = C_1, C_2, C_3, ..., C_n$, where, n is the total number.

Definition 2: There exists a document cluster (C_i) and textual document (D) where, C_i consists of one or several D. Therefore, $C_i = D_1, D_2, D_3, ..., D_m$, where, m is the total number of documents in a cluster.

Definition 3: There also exists a set of words (W) in D_i that can be represented as $D_i = W_1, W_2, W_3....., W_r$, where, W is word/term and r is the number of words.

Definition 4: Some of the words ($W_{selected}$) have repeated occurrence in D_{all} and they can be Frequent Term (FT) or Name Entity (NE). All words that represent the FT is $W_{frequent}$ and words for name entity name is W_{name}. Therefore:

$$W_{frequent} \subset W_{selected} \text{ and } W_{name} \subset W_{selected}$$

$$FT = W_{frequent} = \{W_1, W_2, W_3, ..., W_s\} \text{ and } NE = W_{name} = \{W_1, W_2, W_3, ..., W_t\}$$

where, s and t are the total number of words in the respective sets $FT \cap NE = \emptyset$. In other words:

$$\exists W \in FT \cup NE$$

Definition 5: Some of the frequent term ($W_{frequent}$) have Maximal Frequent Terms (MFT) ($W_{MaxFrequent}$) in textual document (D_i), Therefore:

$$FT = W_{frequent} = MFT = W_{MaxFrequent} = \{W_1, W_2, W_3, ..., W_q\}, \quad W_{MaxFrequent} \subset W_{frequent}$$

where, q is the total number of maximum frequent words in the respective sets.

Definition 6: Based on the Maximal Frequent Terms (MFT) and Name Entity (NE), a cluster description (P) is constructed to describe the actual data cluster. According to definition 1, D_{all} can be clustered into one or more C. Thus, D_{all} has more than one cluster descriptions represented as follows:

$$D_{all} = \{P_1, P_2, P_3, ..., P_y\}$$

where, y is the total number of cluster description.

Based on the definitions the similarity measure is constructed.There are two phases in performing the similarity measure; creating and retrieving data clusters. In creating data clusters, two types of similarity measures are used for document-to-document and document-to-cluster descriptions, in which either one or both FT and NE can be applied. All similarity measures provide S (FT), which is the semantic (synonym) of frequently occurring words. The S (FT) is provided by using the WordNet database.

In measuring similarity of a document-to-document, clustering is executed on a pair of textual documents with minimum support of words (The number of words determine whether textual document belongs to the existing cluster, where the words should exist in both, D_i and D_j/$CDES_j$, this is called minimum support), as indicated in mathematical Eq. 1. The similarity measure between document (D_i) and document (D_j) with all frequent terms can be represented as the following:

$$Sim(D_i, D_j) = (S(TF) \vee NE)_{Di} \cap (S(TF) \vee NE)_{Dj} \leq min\_support \quad (1)$$

The clustering process can be executed in only a maximal number of occurrences in frequently terms to produce good-quality data clusters, as indicated in mathematical Eq. 2. The similarity measure between document (D_i) and document (D_j) with maximal frequent terms can be represented as the following:

$$Sim(D_i, D_j) = (S(TF_{Max}) \vee NE)_{Di} \cap (S(TF_{Max}) \vee NE)_{Dj} \leq min\_support \quad (2)$$

In document-to-cluster description, clustering process is executed on a single textual document with the cluster description that holds the FT and NE that has the minimum support of words, as indicated in mathematical Eq. 3. The similarity measure between documents (D_i) and cluster description (P_j) with all frequent terms can be represented as the following:

$$Sim(D_i, CDES_j) = (S(TF) \vee NE)_{Di} \cap (S(TF) \vee NE)_{Pj} \leq min\_support \quad (3)$$

The clustering process can be executed in only a maximal number of frequently occurring terms to produce good-quality data clusters, as shown in mathematical Eq. 4. The similarity measure between documents (D_i) and cluster description (P_j) with maximal frequent terms can be represented as the following:

$$Sim(D_i, P_j) = (S(TF_{Max}) \vee NE)_{Di} \cap (S(TF_{Max}) \vee NE)_{Pj} \leq min\_support \quad (4)$$

Conversely to creating data clusters, retrieving data clusters can be used when users retrieve textual data. Equation 5 represents the percentage similarity between textual document description and cluster description. The percentage similarity between document and cluster description can be weighed as the following:

$$S\left(D_i, CDES_j\right) = \frac{\sum_{i=1}^{i=v}(FT)D_i + \sum_{i=1}^{i=g}(NE)D_i}{\sum_{j=1}^{j=v,g}(FT \vee NE)CDES_j} \quad (5)$$

where, v, g number are frequent terms and number of named entity respectively.

RESULTS AND DISCUSSION

There are two experiments conducted to evaluate performance and accuracy of textual document clustering for TVSM. The TVSM developed by Oracle 11G and Java

Table 1: Summary of data description

Dataset	Reuters	Classic	WAP
No. of documents	10500	7094	1560
No. of classes	5	4	20
Maximum class size	3880	3203	342
Minimum class size	1570	1033	5
Average class size	2100	1774	78

Development Kit (JDK) 1.6. The experiment is performed on a PC with AMD Phenom ii x4 B93 2.8 GHz and 4 GB RAM operating under windows 7. In order to compare TVSM with the most popular clustering algorithms for document clustering such as k-mean, bisecting k-mean (BKM), agglomerative hierarchical clustering with UPGMA, Oracle Text and frequent term-based hierarchical clustering (FTHC)[21], the RapidMiner[54] tool and CLUTO[55,56] kit tool are used to extract and cluster textual documents from three different datasets.

Dataset: The experiment is conducted on three types of datasets: Reuters news stories (Reuters corpus, volume 1) (RCV1)[57], a classic dataset[58] and Web Application Project (WAP). The RCV1 contains hundreds of thousands of textual documents in many different categories that comprise news articles. The selected collection of textual documents can be classified into six categories: Economics, corporate (industrial), government (social), politics, market and sports. Classic datasets, which contain abstracts of papers are classified into four categories, namely, CACM, CRAN, CISI and MED. The WAP dataset, which contains a variety of news articles consisting of 20 classes. Table 1 shows a summary of all dataset descriptions. The evaluation process of VSM can be viewed from two perspectives: Quality (excellence) of textual data clusters and efficiency (performance).

Cluster quality: Cluster quality is measured by the overall F-measure (F-score)[59], entropy[46,51] and purity[28,46], which are these the standard measures for cluster quality. Equation 6 represented the calculated overall F-measure on the basis of the obtained F-measure from Eq. 7, which is a mix of recall and precision as Eq. 8 and 9. These measures generate accurate results in information retrieval:

$$\text{Overall F}-\text{measure} = \sum_{c=1}^{c=n} \frac{DCn}{D} \text{Max (f}-\text{measure (i, j)} \quad (6)$$

$$\text{F}-\text{measure} = \frac{2 \times \text{Recall (i, j)} \times \text{Presion (j, j)}}{\text{Recall (i, j)} + \text{Presion (i, j)}} \quad (7)$$

$$\text{Recall (i, j)} = \frac{M_{ij}}{N_i} \quad (8)$$

$$\text{Presion (i, j)} = \frac{M_{ij}}{N_i} \quad (9)$$

where, M_{ij} denotes the number of documents of class i in cluster j, N_i denotes the total number of documents in class i and N_j denotes the total number of documents in cluster j. The second measurements that used is entropy. Entropy measure homogeneous and distribution of documents over all data clusters. The total entropy can be calculated from mathematical Eq. 10, ej can be calculated from mathematical Eq. 11:

$$\text{Total entropies} = \sum_{i=1}^{i=k} \frac{m_j}{m} ej \quad (10)$$

$$\text{Entropy} = \sum_{i=1}^{i=c} P(i, j) \log_2 P(i, j) \quad (11)$$

where, $P_{i,j}$ is probability of member of cluster j belong to class i, the entropy is the precision. It can be calculated as the following mathematical Eq. 12:

$$P(i, j) = \frac{m_{ij}}{m_j} \quad (12)$$

The third measures is purity, it assesses the purity of data clusters. It can be calculated by the following mathematical Eq. 13:

$$\text{Overall purity} = \sum_{j=1}^{j=k} \frac{n_j}{m} \text{purity}_j \quad (13)$$

where, n_j is number of documents in cluster j and N denotes total number of documents in all data clusters and purity_j is entropy of cluster j. The maximize F-measure and minimize entropy that shows the best and high quality of data clusters. In addition, the high value of purity is indicates better clustering process, where $\text{Purity}_j = \text{Max}_p (i, j)$.

Based on the mathematical equations the data cluster quality is measured. Table 2 shows the comparison of cluster qualities.

In the performance of data cluster query retrieval experiment, the textual documents are divided into seven groups to differentiate the performance of creating data clusters on each group. These groups contain 1000, 2000, 4000, 6000, 8000, 10,000 and 50,000 textual documents. All the textual data clustering algorithms used in the experiments predefine the number of clusters that is the user should first

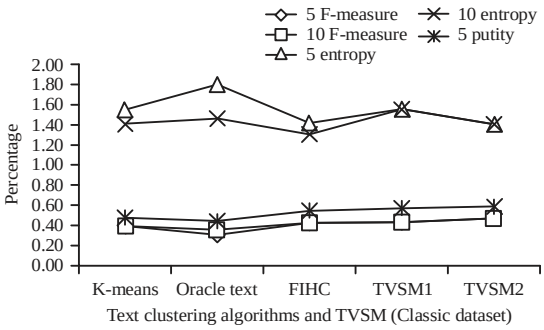

Fig. 4: Quality of data clusters

Table 2: Assessment of data cluster quality

Dataset	Clusters	Measures	K-means	Oracle text	FIHC	TVSM1	TVSM2
Classic	5	F-measure	0.4	0.3	0.43	0.44	0.47
	10	F-measure	0.4	0.36	0.43		
	5	Entropy	1.55	1.81	1.42	1.56	1.41
	10	Entropy	1.41	1.47	1.31		
	5	Purity	0.48	0.45	0.56	0.57	0.59
	10	Purity	0.57	0.56	0.64		
WAP	5	F-measure	0.3	0.24	0.43	0.43	0.51
	10	F-measure	0.3	0.29	0.5		
	5	Entropy	3.2	3.39	2.59	2.55	2.33
	10	Entropy	2.82	2.86	2.33		
	5	Purity	0.31	0.28	0.44	0.49	0.55
	10	Purity	0.37	0.33	0.51		
Reuters	5	F-measure	0.39	0.36	0.4	0.43	0.48
	10	F-measure	0.36	0.39	0.39		
	5	Entropy	1.69	1.73	1.69	1.45	1.3
	10	Entropy	1.54	1.62	1.46		
	5	Purity	0.51	0.5	0.5	0.57	0.61
	10	Purity	0.56	0.54	0.54		

enter the number of clusters. In this study, the predefined number of clusters is 5, 10 and 20. However, TVSM does not require such predefinition. Performance is measured on the basis of the Reuters dataset and the unit of measurement for execution time is seconds. Table 3 shows the performance based on retrieving data clusters which created using TVSM and most common used textual data clustering algorithms.

The result of the experiment shows that the performance of TVSM outperforms K-mean, BKM, UPGMA, Oracle text and FICH clustering algorithms. Due to TVSM works as incremental clustering process, there is no need to re-cluster textual data every time. On the other hand, the goodness (quality of clusters) reach to better score as indicated in Fig. 4 and 5. Specifically, when dataset contains news articles, they often hold named entities. The TVSM has been executed used two similarity measures are document to document that represented by TVSM1 and document to cluster description that represented by TVSM2.

Figure 4 and 5 show the comparison of cluster qualities for the most popular algorithms and the proposed model.

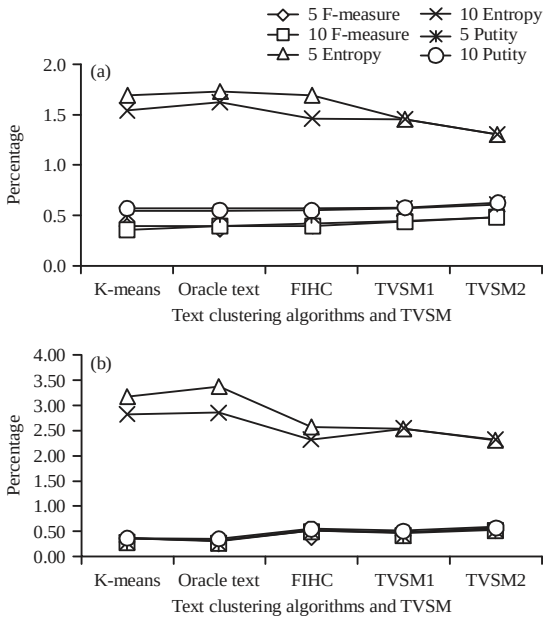

Fig. 5(a-b): Quality of data clusters, (a) Classic dataset and (b) WAP dataset

Table 3: Data cluster query retrieval performance

No. of documents	No. of clusters	K-means	UPGMA	FIHC	Oracle	TVSM
1000	5	23	1800	12	2.34	0.5
	10	24		13	3	
	20	27		13	2.42	
2000	5	63	N/A	13	3.4	0.5
	10	70	N/A	15	4.2	
	20	64	N/A	14	4.88	
4000	5	3.2	N/A	28	5.6	0.5
	10	3.1	N/A	35	7	
	20	3	N/A	28	8.32	
6000	5	5.2	N/A	53	8	0.7
	10	5.15	N/A	53	9.5	
	20	5.5	N/A	52	11.9	
8000	5	11.28	N/A	75	10.2	0.7
	10	11.36	N/A	78	12.4	
	20	15.5	N/A	75	15.58	
10,000	5	19.18	N/A	96	12	0.9
	10	18.2	N/A	97	14.9	
	20	19.52	N/A	99	18.4	
50,000	5	210	N/A	N/A	N/A	0.12
	10	180	N/A	N/A	N/A	
	20	150	N/A	N/A	N/A	

Using the F-measure, entropy and purity, experiments are conducted on the basis of the two proposed similarity measures. In the first similarity, the results show that the TVSM has an F-measure higher than those of BKM and FICH when executing the classic dataset. The second similarity measure achieves a higher F-measure when executed on Reuters news articles due to it clustering based on words that are included in cluster description. Entropy and purity also show that TVSM produces high-quality clusters by generating high entropy and low purity results. All of the experimental measurements

for cluster quality were conducted on three datasets (Reuters WAP and classic) with 1000 documents, which are a mix of all classes The BKM and UPGMA cannot function on large datasets.

CONCLUSION

In this study, we have presented the study of data extraction, unstructured data management and textual clustering on relational DBMS and desktop application. Due to the common phases in the data clustering process that requires tedious coding and scripting. In addition, relational database contains huge unstructured data thus user encounter difficulty in finding useful information. We propose a Textual Virtual Scheme Model (TVSM) for automatic clustering for textual data at column level in relational database. The TVSM will provide quick extraction, data arrangement and grouping for data similar pattern, find linkage between textual documents and improve the query processing performance in relational databases. In addition, converts unstructured information into structured by semantic linking. The system architecture can be applied in many application areas such as topic detection and tracking, textual document clustering, news clustering and web record. The results of experiments showed that quality of data clusters is acceptable when F-measure, entropy and purity are used. In addition, entropy and purity measurement are applied to evaluate quality of data clusters.

SIGNIFICANT STATEMENT

The proposed model focuses on an automatic and incremental strategy to transforms unstructured textual data into structured form. As a result of this strategy, the textual data is automatically represented by the most common words before storing it to database records. By using these common words the clustering assignment process can be achieved incrementally. In clustering assignment processes,the textual document belongs to an existing data cluster. Once the existing textual data cluster is determined, the data are considered as a cluster for the textual document. Otherwise, the process considers creating new textual data clusters. In addition, these common words can be a guide for the next textual document to be stored if the documents bear any similarity with textual data.

ACKNOWLEDGMENT

This study based on work support by Universiti Teknologi MARA (UiTM), Malaysia. The author would like to thanks UiTM.

REFERENCES

1. Doan, A., R. Ramakrishnan, F. Chen, P. DeRose and Y. Lee *et al*., 2006. Community information management. IEEE Data Eng. Bull., 29: 64-72.
2. Chiticariu, L., Y. Li, S. Raghavan and F.R. Reiss, 2010. Enterprise information extraction: recent developments and open challenges. Proceedings of the ACM SIGMOD International Conference on Management of Data, June 6-10, 2010, Indianapolis, IN., USA., pp: 1257-1258.
3. Simoes, G., H. Galhardas and L. Coheur, 2009. Information extraction tasks: A survey. Proceedings of the INForum 2009, May 27-29, 2009, Prague.
4. Muslea, I., 1999. Extraction patterns for information extraction tasks: A survey. Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction, Volume 2, July 18-19, 1999, Orlando, FL., USA.
5. Sarawagi, S., 2008. Information extraction. Foundat. Trends Databases, 1: 261-377.
6. Tari, L., P.H. Tu, J. Hakenberg, Y. Chen, T.C. Son, G. Gonzalez and C. Baral, 2010. Parse tree database for information extraction. IEEE Trans. Knowledge Data Eng.
7. Consens, B., H. Sql, G.E. Blake, M.P. Consens and P. Kilpelainen *et al*., 1994. Text/relational database management systems: Harmonizing SQL and SGML. Proceedings of the 1st International Conference on Applications of Databases, June 21-23, 1994, Vadstena, Sweden, pp: 267-280.
8. Mansuri, I.R. and S. Sarawagi, 2006. Integrating unstructured data into relational databases. Proceedings of the 22nd International Conference on Data Engineering, April 3-7, 2006, Atlanta, GA., USA., pp: 29.
9. Farber, F., S.K. Cha, J. Primsch, C. Bornhovd, S. Sigg and W. Lehner, 2012. SAP hana database: Data management for modern business applications. ACM SIGMOD Rec., 40: 45-51.
10. Li, W. and B. Lang, 2010. A tetrahedral data model for unstructured data management. Sci. China Inform. Sci., 53: 1497-1510.
11. Liu, X., B. Lang, W. Yu, J. Luo and L. Huang, 2011. AUDR: An advanced unstructured data repository. Proceedings of the IEEE 6th International Conference on Pervasive Computing and Applications, October 26-28, 2011, Port Elizabeth, South Africa, pp: 462-469.
12. Kandogan, E., R. Krishnamurthy, S. Raghavan, S. Vaithyanathan and H. Zhu, 2006. Avatar semantic search: A database approach to information retrieval. Proceedings of the ACM SIGMOD International Conference on Management of Data, June 27-29, 2006, Chicago, Illinois, USA., pp: 790-792.
13. Agichtein, E. and L. Gravano, 2003. Querying text databases for efficient information extraction. Proceedings of the 19th International Conference on Data Engineering, March 5-8, 2003, Bangalore, India.

14. Agrawal, S., S. Chaudhuri and G. Das, 2002. DBXplorer: A system for keyword-based search over relational databases. Proceedings of the 18th International Conference on Data Engineering, February 26-March 1, 2002, San Jose, CA., pp: 5-16.
15. Park, J. and S.G. Lee, 2011. Keyword search in relational databases. Knowledge Inform. Syst., 26: 175-193.
16. Guinepain, S. and L. Gruenwald, 2005. Research issues in automatic database clustering. ACM SIGMOD Rec., 34: 33-38.
17. Gupta, V. and G.S. Lehal, 2009. A survey of text mining techniques and applications. J. Emerg. Technol. Web Intell., 1: 60-76.
18. Sahoo, N., J. Callan, R. Krishnan, G. Duncan and R. Padman, 2006. Incremental hierarchical clustering of text documents. Proceedings of the 15th ACM International Conference on Information and Knowledge Management, November 5-11, 2006, Arlington, VA., USA., pp: 357-366.
19. Rani, M.U., 2011. Review on frequent item-based dynamic text clustering. Int. J. Comput. Sci. Inform. Technol. Secur., 1: 98-103.
20. Beil, F., M. Ester and X. Xu, 2002. Frequent term-based text clustering. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002 Edmonton, Alberta, Canada, pp: 436-442.
21. Fung, B.C.M., K. Wangy and M. Ester, 2003. Hierarchical document clustering using frequent itemsets. Proceedings of the 3rd SIAM International Conference on Data Mining, May 1-3, 2003, San Francisco, CA., USA.
22. Chen, C.L., F.S.C. Tseng and T. Liang, 2010. Mining fuzzy frequent itemsets for hierarchical document clustering. Inform. Proc. Manage., 46: 193-211.
23. Liu, X. and P. He, 2005. A Study on Text Clustering Algorithms Based on Frequent Term Sets. In: Advanced Data Mining and Applications, Li, X., S. Wang and Z.Y. Dong (Eds.). Springer, USA., ISBN: 978-3-540-27894-8, pp: 347-354.
24. Malik, H.H. and J.R. Kender, 2006. High quality, efficient hierarchical document clustering using closed interesting itemsets. Proceedings of the 6th International Conference on Data Mining, December 18-22, 2006, Hong Kong, pp: 991-996.
25. Krishna, S.M. and S.D. Bhavani, 2010. An efficient approach for text clustering based on frequent itemsets. Eur. J. Scient. Res., 42: 385-396.
26. Hernandez-Reyes, E., R.A. Garcia-Hernandez, J.A. Carrasco-Ochoa and J.F. Martinez-Trinidad, 2006. Document clustering based on maximal frequent sequences. Proceedings of the 5th International Conference on Advances in Natural Language Processing, August 23-25, 2006, Turku, Finland, pp: 257-267.
27. Zhang, W., T. Yoshida, X. Tang and Q. Wang, 2010. Text clustering using frequent itemsets. Knowledge-Based Syst., 23: 379-388.
28. Li, Y., S.M. Chung and J.D. Holt, 2008. Text document clustering based on frequent word meaning sequences. Data Knowledge Eng., 64: 381-404.
29. Shehata, S., F. Karray and M. Kamel, 2006. Enhancing text clustering using concept-based mining model. Proceedings of the IEEE 6th International Conference on Data Mining, December 18-22, 2006, Hong Kong, China, pp: 1043-1048.
30. Huang, A., D. Milne, E. Frank and I.H. Witten, 2009. Clustering documents using a wikipedia-based concept representation. Adv. Knowledge Discovery Data Min., 5476: 628-636.
31. Baghel, R. and D.R. Dhir, 2010. A frequent concepts based document clustering algorithm. Int. J. Comput. Appl., 4: 6-12.
32. Montalvo, S., V. Fresno and R. Martinez, 2012. NESM: A named entity based proximity measure for multilingual news clustering. Procesamiento Lenguaje Natural, 48: 81-88.
33. Cao, T.H., H.T. Do, D.T. Hong and T.T. Quan, 2008. Fuzzy named entity-based document clustering. Proceedings of the IEEE International Conference on Fuzzy Systems, June 1-6, 2008, Hong Kong, pp: 2028-2034.
34. Cao, T.H., T.M. Tang and C.K. Chau, 2012. Data Mining: Foundations and Intelligent Paradigms. Springer, New York, pp: 267-287.
35. Cutting, D.R., D.R. Karger, J.O. Pedersen and J.W. Tukey, 1992. Scatter/gather: A cluster-based approach to browsing large document collections. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, June 21-24, 1992, Copenhagen, Denmark, pp: 318-329.
36. Gil-Garcia, R., J.M. Badia-Contelles and A. Pons-Porrata, 2006. A general framework for agglomerative hierarchical clustering algorithms. Proceedings of the 18th International Conference on Pattern Recognition, August 20-24, 2006, Hong Kong, pp: 569-572.
37. Gil-Garcia, R. and A. Pons-Porrata, 2010. Dynamic hierarchical algorithms for document clustering. Pattern Recognit. Lett., 31: 469-477.
38. Yafooz, W.M.S., S.Z.Z. Abidin, N. Omar and R.A. Halim, 2014. Model for Automatic Textual Data Clustering in Relational Databases Schema. In: Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013), Herawan, T., M.M. Deris and J. Abawajy (Eds.). Springer, Singapore, ISBN: 9789814585187, pp: 31-40.
39. Yafooz, W.M., S.Z. Abidin and N. Omar, 2011. Towards automatic column-based data object clustering for multilingual databases. Proceedings of the IEEE International Conference on Control System, Computing and Engineering, November 25-27, 2011, Penang, Malaysia, pp: 415-420.
40. Yafooz, W.M., S.Z. Abidin, N. Omar and R.A. Halim, 2013. Dynamic semantic textual document clustering using frequent terms and named entity. Proceedings of the IEEE 3rd International Conference on System Engineering and Technology, August 19-20, 2013, Malaysia, pp: 336-340.

41. Lextek, 2014. Onix text retrieval toolkit-API reference. http://www.lextek.com/manuals/onix/stopwords1.html
42. Porter, M.F., 1980. An algorithm for suffix stripping. Program Electron. Lib. Inform. Syst., 14: 130-137.
43. SNLP., 2013. Stanford Named Entity Recognizer (NER). The Stanford Natural Language Processing. http://nlp.stanford.edu/software/CRF-NER.shtml
44. Finkel, J.R. and C.D. Manning, 2009. Joint parsing and named entity recognition. Proceedings of the Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, May 31-June 5, 2009, Boulder, pp: 326-334.
45. Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM, 38: 39-41.
46. Huang, A., 2008. Similarity measures for text document clustering. Proceedings of the New Zealand Computer Science Research Student Conference, April 14-17, 2008, Christchurch, New Zealand, pp: 49-56.
47. Sharma, A. and R. Dhir, 2009. A wordsets based document clustering algorithm for large datasets. Proceeding of the IEEE International Conference on Methods and Models in Computer Science, December 14-15, 2009, Delhi, pp: 1-7.
48. Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surv., 31: 264-323.
49. Su, C., Q. Chen, X. Wang and X. Meng, 2009. Text clustering approach based on maximal frequent term sets. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 11-14, 2009, San Antonio, TX, USA., pp: 1551-1556.
50. Malik, H.H., J.R. Kender, D. Fradkin and F. Moerchen, 2010. Hierarchical document clustering using local patterns. Data Mining Knowledge Discov., 21: 153-185.
51. Shehata, S., F. Karray and M. Kamel, 2010. An efficient concept-based mining model for enhancing text clustering. IEEE Trans. Knowl. Data Eng., 22: 1360-1371.
52. Montalvo, S., R. Martinez, A. Casillas and V. Fresno, 2007. Bilingual News Clustering using Named Entities and Fuzzy Similarity. In: Text, Speech and Dialogue, Matousek, V. and P. Mautner (Eds.). Springer, Berlin, Heidelberg, ISBN-13: 9783540746270, pp: 107-114.
53. Friburger, N., D. Maurel and A. Giacometti, 2002. Textual similarity based on proper names. Proceedings of 25th ACM SIGIR Conference Workshop Mathematical Formal Information Retrieval, August 11-15, 2002, Finland, pp: 155-167.
54. RipidMiner, 2014. RipidMiner data mining tool. http://rapid-i.com/content/view/181/190/
55. Karypis, G., 2002. CLUTO-a clustering toolkit. https://www.cs.umn.edu/research/technical_reports/view/02-017
56. Zhao, Y., G. Karypis and U. Fayyad, 2005. Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Discov., 10: 141-168.
57. Reuters, 2014. Reuters dataset. http://about.reuters.com/researchandstandards/corpus/
58. Tunali, V., 2010. Classic3 and classic4 datasets. http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/
59. Yang, F., T. Sun and C. Zhang, 2009. An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization. Exp. Syst. Applic., 36: 9847-9852.