International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.6., 2015
(Nov.-Dec.,)

RESEARCH ARTICLE

ISSN: 2321-7758

RULE BASED DATA PURIFICATION (RuBDaP) MODEL FOR BIG DATA ENVIRONMENT

SABITHA M.S[1], DR.S.VIJAYALAKSHMI[2], RATHIKAA SRE R.M[3]

[1]Research Scholar , Research & Development Centre, Bharathiar University, Coimbatore
[2]Assistant Professor, Thiagarajar College of Engineering, Madurai,
[3]Student, Mepco Schlenk Engineering College, Sivakasi

SABITHA M.S

ABSTRACT

Data Purification is one of the significant step in data mining approach. It deals with detection and removal of dirty data from the file. This technique helps to reduce the file size and speed up the analytics tasks. This paper proposes a Rule Based Data Purification (RuBDaP) model. This model works in three stages. The first stage is the incipient phase which assigns rules to various columns. Next phase filters the dirty data and the data purified in the final stage. The experiment proved the quality of the data improved after data purification process.

Keywords—Data preprocessing, Data purification, Big Data

I. INTRODUCTION

In the recent technology era, various developments are happening in online and internet technologies. By this way, it generates huge amount of data from many resources and services which were not available few years ago. Huge amount of data are generated about people and their communications. Various enterprises require potential benefits from the social media like Facebook, Google, Twitter, Linkedin etc., for their business improvement. Cloud storages, social networks and etc., generates huge volume of data. It requires proper management and data analysis. Even though this huge volume of data is useful for the enterprises it may create problems also. Therefore this big data has its own demerits also. The data received from various online technologies are unreliable in nature. The online generated data are suspected to different types of corruptions like wrong, lost and conflict form.

Data is the word used everywhere for everything. The quality of data is very important everywhere that too especially in big data. Data which are found in the real world is incomplete, noisy and dirty. Usage of these data for analytics may lead to various types of errors. Data cleansing is the time consuming step in the data analysis process. Identification and correction of the data errors expects manual review of data in various phases which is a time consuming process.

If such data are used for further analysis eventually the result would be with poor quality. Quality of data is measured in terms of accuracy, completeness, consistency, timeliness, believability, and interpretability. When the user wants to find out the facsimile record, missing data, record & field similarities, duplicate elimination the data pre-processing is used.

The proposed paper is a Rule Based Data Purification (RuBDaP) model. Secion II discusses the views of various researchers. The proposed model

illustrated in Section III and the experiments shows in Section IV. This paper concluded in section V.

II. RELATED WORK

Data pre-processing or data purification is actually rectifying the errors or problems in the database. The views of various researchers are discussed in this section.

a. Zhu Yan-li, Zhang Jia, "Research on Data Preprocessing In Credit Card Consuming Behavior Mining," 2012.

In this paper we study the data preprocessing of credit card system in association analysis, risk detection and customer segmentation. There are so many tables in the system and six tables were chosen from the database related to the topic. After that selection of attributes from each table is done. Then the statistical fields are added. After these data are extracted the cleaning process is done. The cleaning is done as a four step process i) dirty data removal ii) handling null values iii) rectifying the false data iv) removing duplicate data. When removing the dirty data all irrelevant details are removed. Handling null values is done by replacing relevant data into the null fields. The false data are rectified by using query analyzer. The removal of duplicate data is done using the Oracle 'distinct' keyword. In the next step the cleaned data is reduced and integrated. It is done as a tree step the segmentation, association analysis and risk detection. At last the relevant data mining algorithm is selected. This study finally gave results about the customer segmentation model and its application in the credit card industry.

b. Li Chaofeng, "Research and Development of Data Preprocessing in Web Usage Mining,"

In this paper the data preprocessing in Web mining is analyzed to identify the user sessions and identification. The preprocessing is done as a four step process the data cleaning, user identification, session identification, and path completion. The cleaning is done. Accessorial resources embedded in HTML file, robots request and error request were cleaned. The users are identified based on the IP address and logs. Certain rules are framed for this process. The previous method to identify the user session is the timeout and maximal forward reference mechanism. Here after the user identification for the identified users sessions are tracked according to the page request. The rules for the session is whenever new user logs in, when one refer page is null there is a new session, and time between the sessions exceed certain period the new session begins. Finally the path completion is done. All these are proved by conducting an experiment with the web server log of the library of South-Central University for Nationalities. The entries of the raw web log was 747890 and entries after the data cleaning was 112783, number of users 55052 and number of sessions 57245. All these results were graphically represented in the bar graph. The quality of data is increased after this experiment.

c. Jebamalar Tamilselvi, Saravanan, " A Unified Framework and Sequential Data Cleaning Approach for a Data Warehouse," 2008.

In this paper a new framework is proposed for data cleaning and gives a solution to handle data cleaning in a sequential order. The existing approaches for the cleaning like the similarity functions, duplicate elimination function is combined together to perform cleaning in a sequential manner. This framework involves six steps. The attributes are identified and selected. The similarities between records are checked by creating a token. The records are grouped based on the clustering key. Based on the data type the similarity function is selected. For eliminating the duplicates elimination function is created. At last the cleaned data dis merged using the merging technique. A powerful data cleaning tool can be developed using this framework. The token concept speeds up the data cleaning reduces the number of comparisons. The framework can handle all kinds of data in the relational database.

d. Vijay Kumar Padala, Sayeed Yasin,Durga Bhavani Alanka, " A Novel Method for Data Cleaning and User-Session Identification for Web Mining," 2013.

In this paper an algorithm for cleaning a web log file, user and session identification is proposed. Three algorithms are proposed. Initially for cleaning a web log file. The web server log file is taken as input. In the algorithm the raw web log files

are cleaned and insert these proposed data into a relational database. The log database is given as output of the algorithm

In the user identification algorithm the IP address is used for the unique user identification. Whenever a new IP address is used new user is identified. The log database is taken as input and the unique users database is given as output. In the final algorithm the sessions are identified. Whenever the user time exceeds 30 minutes a new session is created. As a result the session database is created. The websites which asks for the user name and password can have log of the users count of visiting the site and the sessions they have used.

e. Ashish R. Jagdale, Kavita V. Sonawane, Shamsuddin S. Khan, "Data Mining and Data Pre-Processing for Big Data," 2014.

In this paper a pre-processing algorithm is proposed to extract real time user accessed data from windows os and Apache HDFS framework using Map Reduce Functionality. There are two phases in the proposed system 1. Data Pre-Processing and 2. Data mining and Analysis. In the data pre-processing the real time data collection is done such as the data from individual user machine accessing different files and folders. Then the log file is created by cleaning and loading the data. Then the data which are related are grouped together to form the dataset. In the data mining analysis the data mining algorithm is applied to the dataset which are accessed in different period of time. The output is analyzed. Using the apache hadoop packages the experiment is done and results are represented graphically. Based on different parameters like execution time, data scalability and flexibility and data heterogeneity the performance analysis of the proposed system is done. Single node is used for this algorithm and in future it can be applied to nodes that are distributed geographically in different locations by parallel running of Map Reduce programming model.

III RULE BASED DATA PURIFICATION (RUBDAP) MODEL

Data purification is a significant element to ameliorate the accuracy of the data analysis. In the purification step, the data are treated in order to employ the analysis process achievable and effective.

The most common elements used in Big Data are nominal, discrete and continuous. Nominal is the usage of limited number of values that do not have any relationship. Continuous type will have the unlimited number of values. But in Big data there may the some relation as well as the limited number of values. This will improve the quality as well as speed of the analysis process. Table 1 represented in Fig 1 is a customer / product table. Each record in this table represents individual information.

	c1	c2	c3	c4	c5	c6	c7	c8	c9
	Cust_id	Cust_name	Region	City	Product	Part_no	Price	Sales	Release
r1	101	Global	East	MD	Switch	S0212	105	1000	Apr 15
r2	102	Techsoft	West	NM	Rod	R0113	125	100	Feb 2015
r3	105	Ultra	South	MA	Sensor	S0102	105	2500	01/2016
r4	102	Vector	North	NM	Sensor	S0012	105	NULL	June 15
r5	106	Gill	East	SA	Pipe	P1045	185	NULL	Jul 15
r6	108	Bost	West	NM	RRod	R0113	105	670	Jun 15
r7	109	Creative	East	SA	SSensor	S0012	105	8900	July 15
r8	110	Delta	East	SA	Rod	R1013	105	2500	04/15
r9	112	Hydra	North	MD	Pipe	P1045	185	800	02/15

Fig 1 : Sales Table

A portion of Sales Table is given in the above table as Fig1. Considering this table as a model, RuBDaP design proposed in this section. Initially the problems in the table were analyzed based on that the solution provided in the RuBDaP design.

Primary data : In the above table the cust_id column looks like a primary key. But the value 102 repeated in r2 and r4.

Relative data : The cust_name, region and product are related to cust_id , city and part_no respectively. Following dirty data observed from the above sample dataset.- r2 & r4 - mismatch in cust_id and cust_name - r1 & r9, r2,r4 & r6 - conflicts in region field. - r3,r4 & r7 - conflicts in part_no field

Datatype mismatch – Different data type used in a single column. In sales column, K and L are used for 1000s and 10000s instead of integer numbers.

The above table is the sample portion of data in a Big Data database. Manual correction is manageable for a small table but this is not possible for a Big Data. This paper proposes a Rule based Data Purification system (RuBDaP).

The general architecture of the proposed system depicted in Fig 2. Basically this architecture contains three stages. First stage is for fixing rules , next stage is to filter the dirty data and final stage is data elimination or correcting dirty data.

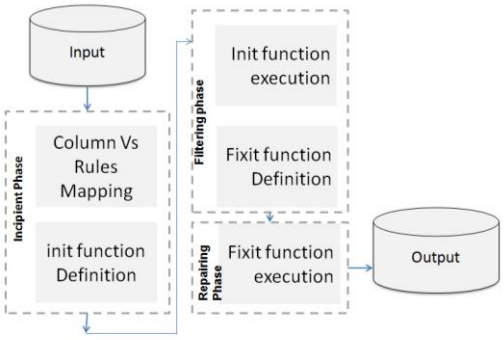

Fig 2. RuBDaP architecture

Incipient Phase

This is the initial phase in the proposed purification model. The rule fixation and locating dirty data covered in the Incipient phase itself. This stage is an important stage. Rules to be assigned for required fields in the table. This system is capable of fixing rules in the following category.

- Unique value assignment
- Referral value substitution
- Data type reclamation
- Harmonizing Data

Framing the rules are an important task in this phase. The proposed system suggests four types of rules. Every columns can undergo one or more rules.

Incipient Phase	Rules
UNIMRK	l1
MAPFLD	l2
FIXTYP	l3
UNIFRM	l4

First step of this process is to assign every column to the various rules recommended in the incipient phase.

Let C = {c1,c2,…,cn} – columns in the table

R = {r1,r2,…,rm} – rows in the table

L = {l1,l2,l3,l4} – rules framed in the incipient phase

Fig 3 illustrates various stages of Incipient phase. The prototype for the calling function in this phase is *Init(<field name>,<Rule name>,<Associated Master name>)*

Field name - column in C.

Rule name – A rule from L

Associated Master name – Master reference data used for reference purpose.

The parameter for the Init function varies depends upon the rules we choose for the operation. For l1 and l3, no need to specify the master name. Appropriate master name selection is required for other rules.

Fig 3. Stages of Incipient Phase

Fig 4 demonstrates various init functions derived for c1,c2…cn available in Table1. UNIMRK and UNIFRM rule is used only for the cust_id and Release respectively. MAPFLD is used for the columns cust_id&cust_name, city&region and product&part_no. FIXTYP is used for the columns Price and Sales. Single column can opt for more than one Rule.

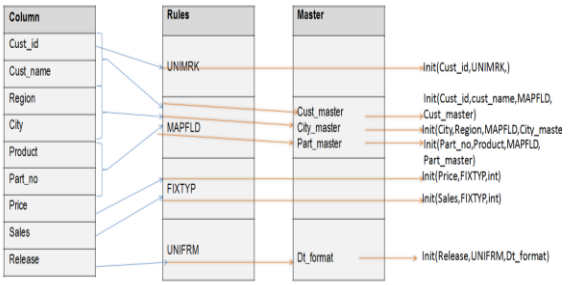

Fig 4. Stages of Incipient Phase for Table 1

Rules used in Incipient Phases are explained below.

UNIMRK : This rule is used to specify the column for unique values. Following method is used to identify the dirty data ie duplicates from the table list.

init(<field_name>,UNIMRK,)

MAPFLD : Most of the dirty data will be removed when applying this rule. This rule associated with various masters. Considering the masters are developed or already available for reference. Selection of appropriate master table and field names will help to remove the dirty data in the table. Providing correct list of columns will yield good results. The init function for this rule will have list of fields indicated as FList and the master_table is the related master table where it has the reference fields.

init(<FList>,MAPFLD,<master_table>)

FList ϵ C

Flist – {f1,f2,f3,….fn}> n – no of fields.

FList is list of fields selected for cross verification.

FIXTYP : Aggregate functions in analysis require proper values related to datatype. For eg., the null value in the numeric datatype may lead to some error in aggregate function.

init(<field_name>,DATTYP,<data type>)

UNIFRM : This rule is to covert different values in a column to equable form. For Eg. Jan/January/01 means the only month January. The syntax to adopt this rule is

init(<field_name>,UNIFRM,<master_table>)

The master table is a table which keeps the equable form of the column data.

In the Incipient Phase the rules fixed for various columns and dirty data identified for purification. The next stage is the filtering phase.

Filtering phase

This phase is to list down all the columns which are having dirty data. In it functions are the input for this phase. Two types of decision can be made in this phase. Either eliminating the unwanted row or amending the rows for further usage. Most of the time deletion is a kind of manual decision. This section explains various functions and methods involved in the amending process. Filtering phase received init functions for further processing. Here also further processing done only based on the rules selected.

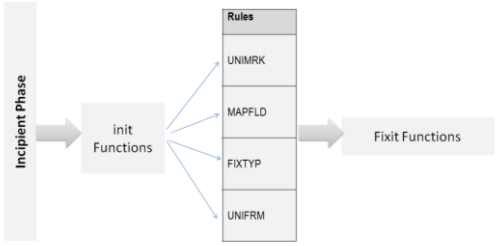

Fig 5. Filtering Phase

The init function received from the first phase is process for producing various Fixit functions. The general format of init is

Init(<field name>,<Rule name>,<Associated Master name>)

The selection statement varies depends on the *<Rule name>* specified in the function.

UNIMRK

Select count<fld_name> from <table> group by <fld_name> having count<fld_name> > 1

Results the duplicate rows <d1,….dx> from the list <r1,r2,…rm) . x is number of duplicates ie dirty data in the table. After getting the duplicates from the list, user intervention required to take decision on amending or deleting the row.The system will call another init statement for amendments

Init(<FList>,MAPFLD,<master_name>)

Fixit(<FList>,MAPFLD,<master_name>)

MAPFLD

Select <FList>,count(*) from <table> group by <FList> having count(*) > 1

Fixit(<FList>,MAPFLD,<master_name>)

FIXTYP

The select statement differs based on the type of the data type we like to change

If float /int /double/numeric
Select case when try_convert(<field_name>,<data type>) is null then <field_name>
Fixit(<field name>,MAPFLD,)
It returns the null value rows from the list.

UNIFRM:

*Select * from <table> where <field_name> not in (select <field_name> from <master_table>)*
Fixit(<field name>,UNIFRM,)

At the end of this phase decision can be made on removal of unwanted rows or repairing the problematic rows. The amenenment / rectifaction addressed in the following phase.

Repairing Phase

This is the final phase of this RuBDaP model. The final phase is explained in the following diagram. This phase executes the Fixit functions. The repairing method varies depends on the type of rule.

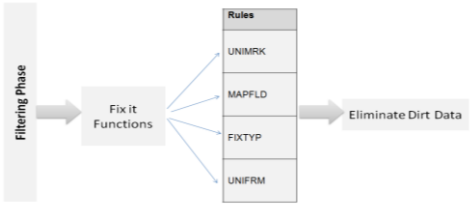

Fig 6. Repairing Phase

The repairing methods based on rules explained below.

UNIMRK

Update <table> set <f2=master_table.f2>, <f3=master_name.f3> ,….. where <f1=master_table.f1>

MAPFLD

Update <table> set <f2=master_table.f2>, <f3=master_name.f3> ,….. where <f1=master_table.f1>

FIXTYP

Update <table> set <field_name=value> where <f1=master_table.f1>
Value – Any value which we like to assign.

UNIFRM

Update <table> set <field_name=master_table.field_name> where <field_name=master_table.field_name>

Now most of the dirty data removed and this purification will speed up the performance of the output.

Fig 7 represents the output of the Table 1(Fig 1) after the purification process using the RuBDaP model.

	c1	c2	c3	c4	c5	c6	c7	c8	c9
	Cust_id	Cust_name	Region	City	Product	Part_no	Price	Sales	Release
r1	101	Global	East	MD	Switch	S0212	105	1000	Apr 15
r2	102	Techsoft	West	NM	Rod	R0113	125	100	Feb 15
r3	105	Ultra	South	MA	Sensor	S0012	105	2500	Jan 16
r4	103	Vector	West	NM	Sensor	S0012	105	0	Jun 15
r5	106	Gill	East	SA	Pipe	P1045	185	0	Jul 15
r6	108	Bost	West	NM	Rod	R0113	105	670	Jun 15
r7	109	Creative	East	SA	Sensor	S0012	105	8900	Jul 15
r8	110	Delta	East	SA	Rod	R1013	105	2500	Apr 15
r9	112	Hydra	East	MD	Pipe	P1045	185	800	Feb 15

Fig 7. Results of Table 1 after purification process

IV Performance Analysis

The performance of the proposed model is discussed in this section. The execution time of the data before and after the purification process measured as a performance measure. The output of the table provides meaningful values before and after the purification process. Different types of analysis explained in various graphs.

City wise/Region wise Sales – The unpurified table not able to create graph due to the data type contains null function. This has been rectified while mapping the with the rule FIXTYP. Fig

8 is the graph created after executing the RuBDaP mechanism.

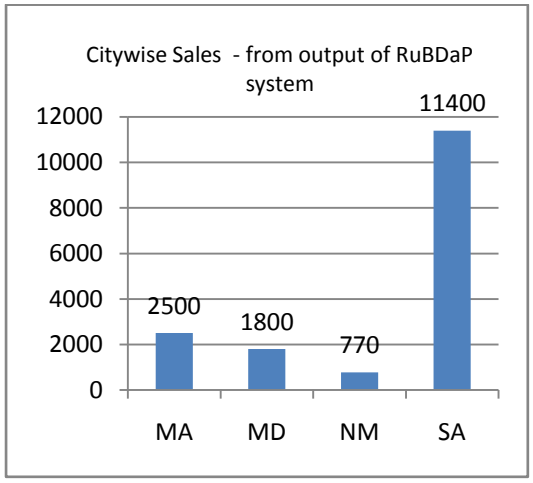

Fig 8. Citywise sales – after RuBDaP System

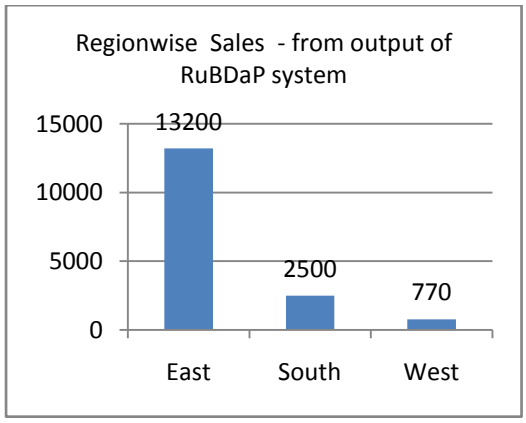

Fig 9. Regionwise sales – after RuBDaP System

Release date wise – Fig 10 and Fig 11 shows number of release in every month. Clarity emerges in Fig 11. After purification process the same type of values modified into a single item. So that the analytics become more meaningful.

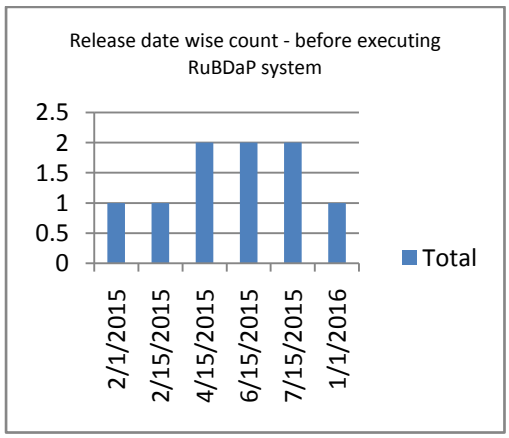

Fig 10. Number releases before purification

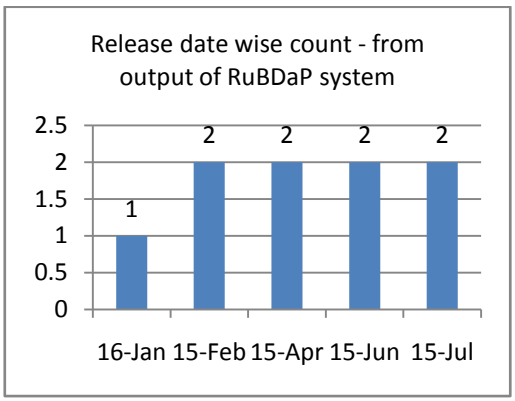

Fig 11. Number releases from purified table.

The analytics performance is not only limited to the above types. It also covers all the columns and helps to provide valuable insights. The performance is not only limited to the output. The execution time also increased due to clarity emerged in the purified data.

CONCLUSION

RuBDaP is a model for speedier and scalable big data purification. It provides user friendly and ease of use programming interface. Users define the rules by mapping with columns in the incipient phase. Then it is transformed to filtering phase to select dirty data. In the final phase all the dirty data converted into relevant values with the help of master tables provided in the system itself for easy reference. But master table is not required in all the cases. This decision made on removal of unwanted rows and amending the existing rows for clarity , performance improvement and for data correctness.

The experiments expressed the advantage of RuBDaP system that the performance increased without sacrificing the quality of the data. Moreover Big RuBDaP is scalable

There are various future research directions in the Data preprocessing model. Introducing more number of logics leads to quality data model.

References

[1]. X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering, ,* vol. 26, pp. 97-107, 2014.

[2]. L. Ismail, M. M. Masud, and L. Khan, "FSBD: A Framework for Scheduling of Big Data

Mining in Cloud Computing," in *IEEE International Congress on Big Data (BigData Congress), 2014*, pp. 514-521.

[3]. A. B. Patel, M. Birla, and U. Nair, "Addressing big data problem using Hadoop and Map Reduce," in *Nirma University International Conference on Engineering (NUiCONE), 2012*, pp. 1-5.

[4]. C. K.-S. Leung and Y. Hayduk, "Mining frequent patterns from uncertain data with MapReduce for Big Data analytics," in *Database Systems for Advanced Applications*, 2013, pp. 440-455.

[5]. B. Lu and S. Wei, "One More Efficient Parallel Initialization Algorithm of K-Means with MapReduce," in *Proceedings of the 4th International Conference on Computer Engineering and Networks*, 2015, pp. 845-852.

[6]. Zhu Yan-li, Zhang Jia, *"Research on Data Preprocessing In Credit Card Consuming Behavior Mining,"* 2012.

[7]. Li Chaofeng, "*Research and Development of Data Preprocessing in Web Usage Mining,"*

[8]. Jebamalar Tamilselvi, Saravanan, " *A Unified Framework and Sequential Data Cleaning Approach for a Data Warehouse,"* 2008.

[9]. Vijay Kumar Padala, Sayeed Yasin,Durga Bhavani Alanka, " *A Novel Method for Data Cleaning and User-Session Identification for Web Mining,"* 2013.

[10]. Ashish R. Jagdale, Kavita V. Sonawane, Shamsuddin S. Khan, "*Data Mining and Data Pre-Processing for Big Data,"* 2014.

[11]. V. López, S. del Río, J. M. Benítez, and F. Herrera, "Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data," *Fuzzy Sets and Systems,* vol. 258, pp. 5-38, 2015.

[12]. S. del Río, V. López, J. M. Benítez, and F. Herrera, "On the use of MapReduce for imbalanced big data using Random Forest," *Information Sciences,* vol. 285, pp. 112-137, 2014.

[13]. J. Evermann and G. Assadipour, "Big data meets process mining: Implementing the alpha algorithm with map-reduce," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, 2014, pp. 1414-1416.

[14]. H. Chai, G. Wu, and Y. Zhao, "A document-based data warehousing approach for large scale data mining," in *Pervasive Computing and the Networked World*, ed: Springer, 2013, pp. 69-81.