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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

A Context Aware Cloud Based Text Mining Recommendation Framework for Vehicle Accidents Using BIME

L.Mahalakshmi, S.Karthik

PG Scholar, Dept. of CSE, Ganadipathy Tulsi's Jain Engineering College, Vellore, Tamil Nadu, India.

Assistant Professor & HOD, Dept. of CSE, Ganadipathy Tulsi's Jain Engineering College, Vellore, Tamil Nadu, India.

ABSTRACT: The accidents on the day to day incidents play a major role in terms of safety concern for the transportation industry. The statistics of road, rail, and air accidents incur a huge loss to the country. Though most of the minor accidents cost very little, it impacts the life of several humans during the time. In order to better understand the accident reports involved, this application is developed on the backend process of text mining where the Apriori is applied using the BIME tool. The Extensive analysis is collecting the data sets of the incidents happened on various factors. It predicts the accuracy on contribution of those accidents and the steps for avoidance. This application assesses the adequacy of content mining of mishap accounts by surveying prescient execution for the expenses of outrageous mischances. The outcomes demonstrate that prescient exactness for mishap costs altogether enhances using highlights found by content mining and prescient precision additionally enhances using cutting edge outfit strategies. Importantly, this study also shows through case examples how the findings from text mining of the narratives can improve understanding of the contributors to accidents in ways not possible through only fixed field analysis of the accident reports. The BIME tool is used on top of BigQuery Table to discover the data analytics of the accidents and its lose. This application overcomes the existing methodologies of manual prediction about the causes and losses incurred.

KEYWORDS: BIME tool, Big Query, Extensive analysis.

I. INTRODUCTION

Text mining is discovery of new and previously unknown information automatically from different text resources using natural language and computation linguistics, machine learning and information science methods [1]. The key element is linking of the discovered information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation. Text mining methods include the steps of processing the input text data, deriving rules and patterns within the newly processed data and finally the evaluation and interpretation of the output rules and patterns [2].

Accidents analysis deals with collection of raw data and evolving the statistic report on the analysis. Taking this in to consideration, the data set consuming the list of accidents and its mode is generated in this application. The generated dataset is fed to Apriori algorithm which deals with the mining of records. The efficacy of text mining helps us on coming down to the exact cost and humans losses involved on that particular accident.

Information can be extracted from lexical co-occurrence, using combinations of different measurement formulae and nonlinear learning algorithms. Of course, much detailed grammatical information cannot be obtained using methods that discard word ordering within sentences, but it is apparent that there is an abundance of rich and complex information that can be extracted by means such as Leximancer. For rapid human appreciation of the information contained within nontrivial amounts of natural language, perhaps the challenge is to choose what level of detail to abstract. Since 1975 the Federal Railroad Administration (FRA) has collected data to understand and find ways to

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

reduce the numbers and severity of these accidents. The FRA has set "an ultimate goal of zero tolerance for rail-related accidents, injuries, and fatalities".

A review of the data collected by the FRA shows a variety of accident types from derailments to truncheon bar entanglements. Most of the accidents are not serious; since, they cause little damage and no injuries. However, there are some that cause over 1M in damages, deaths of crew and passengers, and many injuries. The problem is to understand the characteristics of these accidents that may inform both system design and policies to improve safety [1].

BIME (Business Intelligence Made Easy) makes it easy to discover the data you and your business care about. Anyone and everyone can be an analyst with BIME. Easily create datasets and create custom metrics, reports and dashboards - no SQL required.

II. PRELIMINARY

In this section, the techniques that are already existing concepts of text mining, BIME, Bigquery and Apriori algorithm.

A. Text Mining:

The techniques use from text mining to derive from ensemble methods that combine the results from many models or learners to produce a consensus prediction. The two types of ensembles: boosting and bootstrap aggregation or bagging. Bagging or bootstrap aggregation builds multiple models by resampling subsets of the original data. The subsets are created by randomly removing some of the original predictor variables [1].

Again the estimates from the resulting models are combined to produce one overall estimate. Boosting provides an iterative approach to combining the outputs from a sequence of simple or weak learners to generate a more accurate combined estimate. The bagging method we use in this work is random forests and it again combines results from multiple classification trees [15]. The random forest algorithm builds a group or forest of tree-based models where each tree uses a subset of the predictor variables with a resampling of the training data.

Text mining is concerned with finding patterns in unstructured text. This field has become increasingly important because of the large amounts of data available in documents, news articles, research papers, and accident reports. In many cases text databases are semistructured because in addition to the free text they also contain structured fields that have the titles, authors, dates, and other meta data. The accident reports used in this paper are semistructured. The key goals of text mining are to characterize the Contents of the documents through pattern discovery. These Patterns may then be used for improved information retrieval or, for input into predictive models.

In any case of a definitive objective, most content mining starts with vector space models where reports are spoken to by term-record lattices. These grids have terms as headers for the columns and archives as headers for the segments. The qualities in the cells give the check or frequencies of a term (push) in an archive (segment).

B. Apriori Algorithm:

The General Process - Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent itemsets in a database.
2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

While the second step is straight forward, the first step needs more attention. Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over I and has size $2n - 1$ (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items n in I, efficient search is possible using the downward-closure property of support (also called anti-monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets.

Apriori Algorithm Pseudocode:

procedure **Apriori** (T, minSupport) { //T is the database and min Support is the minimum support L_1= {frequent items};

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

for (k= 2; L_{k-1} !=∅; k++) {
C_k= candidates generated from L_{k-1}
//that is cartesian product L_{k-1} x L_{k-1} and eliminating any k-1 size itemset that is not
//frequent
for each transaction **t** in database **do**{
#increment the count of all candidates in C_k that are contained in t
L_k = candidates in C_k with minSupport }
//end for each
}
//end for
return $\cup_k L_k$;
}

As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k − 1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates. Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all 2 | S | − 1 of its proper subsets [9].

C. Bigquery:

Storing and querying massive datasets can be time consuming and expensive without the right hardware and infrastructure. Google BigQuery is an enterprise data warehouse that solves this problem by enabling super-fast SQL queries using the processing power of Google's infrastructure. Simply move your data into BigQuery and let us handle the hard work. You can control access to both the project and your data based on your business needs, such as giving others the ability to view or query your data.

We can access BigQuery by using a web UI or a command-line tool, or by making calls to the BigQuery REST APIusing a variety of client libraries such as Java, .NET, or Python. There are also a variety of third-party tools that you can use to interact with BigQuery, such as visualizing the data or loading the data.

Tables contain your data in BigQuery. Each table has a schema that describes field names, types, and other information. BigQuery supports the following table types: Native tables: tables backed by native BigQuery storage. External tables: tables backed by storage external to BigQuery. For more information, see Creating and Querying External Data Sources. Views: virtual tables defined by a SQL query. For more information, see Creating views.

Datasets enable you to organize and control access to your tables. A table must belong to a dataset, so you'll need to create at least one dataset before loading data into BigQuery.

BigQuery manages the technical aspects of storing your structured data, including compression, encryption, replication, performance tuning, and scaling. BigQuery stores data in the Capacitor columnar data format, and offers the standard database concepts of tables, partitions, columns, and rows. BigQuery also supports querying data that’s not in BigQuery storage.

D. BIME:

BIME is primarily an analytical tool, not a database. Uploading data can ease collaboration and boost performance but is not actually necessary. When making a basic connection you can opt out of using a BIME storage option. The data will then be kept in a local memory or in the original data source. Alternatively, you can use a Bime storage option that will take a snapshot of the data that is uploaded to the cloud.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

BIME account is password-protected and safe from prying eyes. If a login / password is not enough, we also offer single sign on (SSO) with all the standards of the market (used by Google, Database.com, Facebook etc.), i.e. OAUTH, OPENID and SAML. So you can log into BIME from your Google Apps account, or use the active directory of your company without requiring a login / password. We can even deactivate the use of login/password to delegate the identity management to a third party.

III. PROPOSED ALGORITHM

In the proposed work, we develop a new framework for text mining,that can be integrated to any platform and used for text mining just by providing the data sets as input. The developed framework is built using the Apriori algorithm with combination of BIME API, where we completely use Google BigQuery for datasets parsing. Figure 1 shows the overall process carried out in the proposed work.

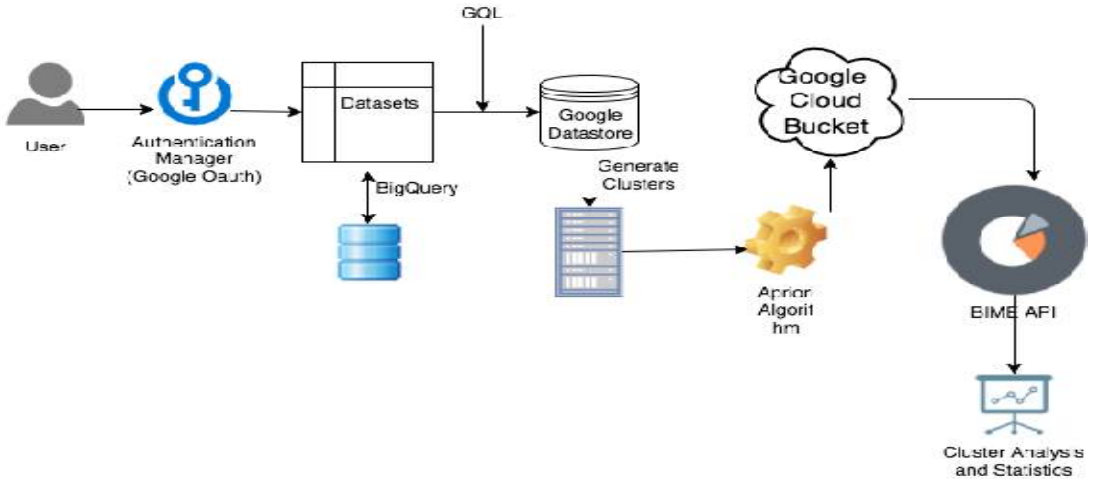

Figure 1.System Architecture

A. *Open Authentication:*

In general, this module deals with authenticating the user inside the application, this is considered to be the efficient and secured method in order to allow the user to be authenticated inside the application. An open authentication protocol is an open standard to authorization. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials.

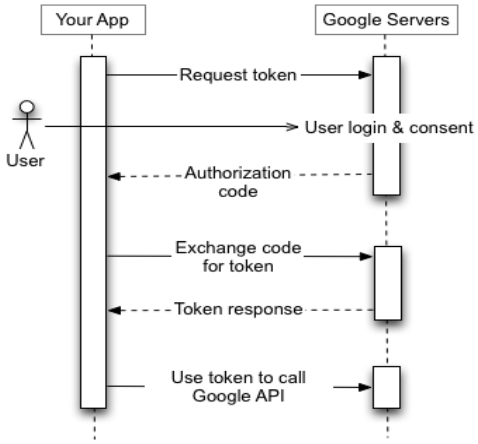

Figure 2.Open Authentication

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner, or end-user. The client then uses the access token to access the protected resources hosted by the resource server. This Figure 2 includes the following steps.

Steps involved in OAuth:

1. Obtain OAuth 2.0 credentials

Both Service Provider (SP) like Google and the application know OAuth 2.0 credentials such as a client ID and client secret. The set of values varies based on what type of application you are building. For example, a JavaScript application does not require a secret, but a web server application does.

2. Obtain an access token from the Google Authorization Server

Before your application can access private data using a Google API, it must obtain an access token that grants access to that API. A single access token can grant varying degrees of access to multiple APIs. A variable parameter called scope controls the set of resources and operations that an access token permits. During the access-token request, your application sends one or more values in the scope parameter. There are several ways to make this request, and they vary by type of application you are building.

For example, a JavaScript application might request an access token using a browser redirect to Google, while an application installed on a device that has no browser uses web service requests. Some requests require an authentication step where the user logs in with their Google account. After logging in, the user is asked whether they are willing to grant the permissions that your application is requesting. This process is called user consent. If the user grants the permission, the Google Authorization Server sends your application an access token (or an authorization code that your application can use to obtain an access token). If the user does not grant the permission, the server returns an error.

3. Send the access token to an API

After an application obtains an access token, it sends the token to a Google API in an HTTP authorization header. It is possible to send tokens as URI query-string parameters, but we don't recommend it, because URI parameters can end up in log files that are not completely secure. Also, it is good REST practice to avoid creating unnecessary URI parameter names.

Access tokens are valid only for the set of operations and resources described in the scope of the token request.

4. Refresh the access token, if necessary

Access tokens have limited lifetimes. If your application needs access to a Google API beyond the lifetime of a single access token, it can obtain a refresh token. A refresh token allows your application to obtain new access tokens. Final after getting the token from google, the key is exchanged between the Google server and the web application, then if the results matches, then the user is allowed to enter in to the application.

B. Datasets Collection and Maintenance:

1. Data Auditing Server:

Once the registration process is over, the API ID and the user details are sent to Auditing agent and the co-users. The notification is sent via the mail API. The mail API requires the following

The Gmail API is a RESTful API that can be used to access Gmail mailboxes and send mail. For most web applications (including mobile apps), the Gmail API is the best choice for authorized access to a user's Gmail data.The Gmail API gives you flexible, RESTful access to the user's inbox, with a natural interface to Threads, Messages, Labels, Drafts, and History. From the modern language of your choice, your app can use the API to add Gmail features like: Read messages from Gmail, Send email messages, Modify the labels applied to messages and threads, Search for specific messages and threads.All you need to use the Gmail API is the client library for your choice of language and an app that can authenticate as a Gmail user. Auditing server analyses the inputs and outputs of the API and mainly checks for the vulnerability issues present on the API.

2. Data Tag Generation:

The tag generation process includes the 256 bit key encrypted tag generated for the API. Initially once the auditing server authorized the API, the next immediate stage, the API file which is uploaded by the user is splitted in to

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

different chunks, where each chunk is associated with the unique reference ID. Apart from the unique chunk ID, the API is possess the separate entity for Tag ID, which is carried out in this phase. The Tag generation process is done via hashing.

3. Push Cloud Datastore:

The datastore is defined as the cloud database server where it persists the data scaled across the globe. Google Cloud Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development.

4. Data Download API:

The download option is set to the users who wants to use the API which they doesn't own/ the API's which other user have uploaded, in order to download the API or access it, the provision is set in such a way that, the downloading user has to provide the API ID and the Tag ID of the API File. On successful verification of the API ID and the Tag ID, the API is downloaded to the mobile. In case of trying for illegal access, unauthorized API access notification is sent to the owner of the API via mail.

C. Pushing Data to Bigquery:

The data can be pushed in many different ways as follows:

- Load from Google Cloud Storage, including CSV, JSON (newline-detimited), and Avro files, as well as Google Cloud Datastore backups.
- Load directly from a readable data source.
- Insert individual records using streaming inserts.

Command Line to Insert:

bq load --source_format = NEWLINE_DELIMITED_JSON[DATASET] [TABLE_NAME] [PATH_TO_SOURCE] [SCHEMA]

D. Generating Cluster Data and Statstical Report:

The first question is important because there is no existing study of the automated use of narrative text for understanding accidents. Our goal is to use predictive accuracy as a metric in assessing the efficacy of using text and text mining to understand contributors to accident damage. The dataset are collected and stored in Google Cloud Storage Bucket. Then perform the BigQuery analysis and query result will be stored in Database.

BigQuery has the success rate of processing 1 Million rows of data sets in 3 seconds. Hence the Apriori algorithm is built using BigQuery in this application for faster process and accurate efficiency. Finally the generated results are fed to BIME API, for statistical analysis of data. The output of the application (Considering 1 Million rows) is expected to be obtained in less than 6 milliseconds which no other framework in this world will give.

IV. CONCLUSION

The In the proposed system, we develop a new reusable generic tool for text mining, which acts as a very powerful framework for all text based semantic analysis. Here, we take road accidents strategy for analysing and statistical purpose to demonstrate the tool. The framework is finally compressed to a JAR file, where it can be reused on java platform as library.

REFERENCES

1. Donald E. Brown, Fellow, IEEE " Text Mining the Contributors to Rail Accidents" 2015
2. Richi Nayak, Noppadol Piyatrapoomi and Justin Weligamage, "Application Of Text Mining In Analysing Road Crashes For Road Asset Management"2009
3. Jianping Cao, Ke Zeng, Hui Wang, Jiajun Cheng, Fengcai Qiao, Ding Wen and Yanqing Gao, "Web-Based Traffic Sentiment Analysis: Methods and Applications" 2014
4. Jianping Cao, Ke Zeng, Hui Wang, Jiajun Cheng, Fengcai Qiao, Ding Wen and Yanqing Gao, "Web-Based Traffic Sentiment Analysis: Methods and Applications" 2014
5. Andrew E. Smith and Michael S. Humphreys, "Evaluation of unsupervised semantic mapping of natural language with Leximancer concept mapping" 2006

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

6. Eleonora D'Andrea, Pietro Ducange, Beatrice Lazzerini, and Francesco Marcelloni, "Real-Time Detection of Traffic From Twitter Stream Analysis" 2015
7. ZHAO Yang, XU Tian-hua, Wang Hai-feng, "Text Mining Based Fault Diagnosis of Vehicle On-board Equipment for High Speed Railway" 2014
8. "Railroad safety statistics—2009 Annual report—Final," Federal Railroad Admin., Washington, DC, USA, Apr. 2011. [Online]Available:http://safetydata.fra.dot.gov/OfficeofSafety/publicsite/Publications.aspx
9. http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Labs-Apriori.pdf
10. L.S. Tey, G. Wallis, S. Cloete, and L. Ferreira, "Modelling driver behaviour towards innovative warning devices at railway level crossings," Neural Comput. Appl., vol. 51, pp. 104–111, Mar. 2013
11. G. Cirovic and D. Pamucar, "Decision support model for prioritizing railway level crossings for safety improvements: Application of the adaptive neuro-fuzzy system," Expert Syst. Appl., vol. 40, pp. 2208–2223, 2013
12. "Office of safety analysis," Federal Railroad Administration, Washington, DC, USA, Oct. 2009. [Online]. Available: http://safetydata.fra.dot.gov/officeofsafety/
13. L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees. Belmont, CA, USA: Wadsworth, 1984
14. T. Hastie, R. Tibshirani, and J. H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer-Verlag, 2009
15. L. Breiman, "Random forests," Mach. Learn., 2001
16. W. Jin, R. K. Srihari, H. H. Ho, and X. Wu, "Improving knowledge discovery in document collections through combining text retrieval and link analysis techniques," in Proc. 7th IEEE Int. Conf. Data Mining, Omaha, NE, USA, Oct. 2007, pp. 193–202

BIOGRAPHY

MAHALAKSHMI.L is a PG Scholar in the Computer Science Department, Ganadipathy Tulsi's Jain Engineering College, Vellore, Tamil Nadu. She received a B.E (CSE) degree in 2015 from under Anna University, Adhiparasakthi College of Engineering, Kalavai, Tamil Nadu. Her research interests are Computer Networks, Neural Networks, MANET etc.

KARTHIK.S is Assistant Professor and HOD in Computer Science and Engineering Department in Ganadipathy Tulsi's Jain Engineering College, Vellore, Tamil Nadu, India.