REAL-TIME ROOT CAUSE IDENTIFICATION ON STREAMING HETEROGENEOUS DATA USING SPARK

S. Charles Britto[1], S.P. Victor[2]

[1]*Research Scholar, Bharathiar University, Tamilnadu, India*
[2]*Associate Professor, Department of Computer Science, St Xavier's College, Tamilnadu, India*

Abstract

Root cause identification can provide huge breakthroughs in the process of business decision making. However, the huge size of input data requires a very highly sophisticated system to perform the analysis. This paper presents a parallelized root cause identification architecture that identifies a ranked list of root causes for the user's query. The major components of the architecture includes conflict identification and elimination, parallelized data pre-processing, polarity identification and polarity based root cause identification. Experiments were conducted on dynamically retrieved data using APIs and the retrieved data is passed to the parallelized components implemented in Spark architecture. The in-memory processing architecture of Spark provides effective results in real-time, making the architecture scalable and robust.

Keywords: *Root Cause Analysis, Sentiment Analysis, Big Data, Spark, Parallelization*

--***--

1. INTRODUCTION

Business analytics has taken a huge leap with the advent of social networks and blogs. Social networks and blogs tend to reflect the real intensions of the users, unlike the conventional feedback forms provided to users. However, identifying appropriate content from the huge storehouse of information is a complex task. Further, such analysis are also hindered by the temporal nature of the data. Old data becomes outdated, however, it still remains and user queries do fetch such data. Hence identifying the freshness of the data becomes mandatory. Further, such data are not free from conflicts. Reviews might provide contradicting opinions, which cannot be appropriately processed by the automated information processing units. Hence it becomes mandatory to retain appropriate information and eliminate the inappropriate content for effective root-cause analysis. The next requirement is to analyze the data and identify the polarity of the data. This can be further extended to identify the opinions or sentiments related to the data [1]. Sentiment analysis or opinion mining is the process of identifying a user's view on an entity by analyzing their textual opinions [2]. This is usually keyword based. User enters their keywords and the appropriate contents are fetched. Analyzing the fetched results provides the user's sentiment towards the entity and also the magnitude of the sentiment level [3]. Some applications of sentiment analysis includes business decision making [4], root cause analysis, recommender systems, clinical systems [5], financial market prediction [6] etc.

All these processes are to be performed on a huge data. Another major challenge is the unpredictable velocity levels exhibited by the social networking data. These challenges warrant the need for Big Data based storage and processing techniques. This paper uses Spark based processing on data retrieved using Web APIs. Conflict identification and resolution is performed on the retrieved data, followed by polarity identification and polarity based root cause identification.

2. RELATED WORKS

Root cause analysis is a recent process, however, the basic components that make up a root cause identification system such as polarity prediction and sentiment analysis or opinion mining have been researched and several contributions can be found in this area. This section describes some of the recent contributions in the area of sentiment analysis, opinion mining and root cause analysis.

An ensemble based sentiment analysis technique was proposed by Aldogan et al. in [7]. This technique uses active learning as the major component of the analysis process. This technique also utilizes two ensemble approaches; probabilistic algorithm and Behavior Knowledge Space (BKS). A phrase based opinion mining technique was presented by Rathi et al. in [8]. This technique uses a phrase based analysis component rather than term or token based analysis to provide effective results. A Twitter based opinion mining technique that measures the service quality of an organization was presented by Takahashi et al. in [9]. This technique operates solely on the basis of customer value, hence providing an additional component for providing value to a review. A machine learning based model that creates word vectors for training the model was presented by Giatsoglou et al. in [10]. A similar vector space based analysis technique that operates on word estimation was proposed by Mikolov et al. in [11], and a similar such vector based estimation model was proposed by Pennington et al. in [12]. Though these technique appear to be effective, the major downside of machine learning based text analysis techniques is that they exhibit high computational complexity levels, hence lack in terms of scaling. A two layer selection algorithm that performs product recommendations to customers was proposed by Li et al. in [13]. Several such recommender techniques operating on user reviews include, a content based recommendation system proposed in [14,15] and a knowledge based system proposed in [16]. An unsupervised sentiment analysis technique to perform multilingual sentiment analysis was presented by Vilares et al. in [17]. This technique utilizes compositional syntax based rules to perform opinion mining.

3. REAL-TIME ROOT CAUSE IDENTIFICATION ON STREAMING HETEROGENEOUS DATA USING SPARK

Identifying root causes is one of the major requirements of the current business. However, the process of identification requires analysis of several huge heterogeneous data sources for the base information. Due to the heterogeneous nature of the data, the inputs are to be processed independently according to their data format. Further, social networking data serves as the major contributor to this process. This leads to the requirement of a real-time processing system. The social network data is usually generated as streams with varied velocity. Hence the algorithm developed for processing such data should also be scalable. This leads to the requirement of Big Data based storage and processing techniques, and Map Reduce based operations. The proposed architecture uses Spark architecture for processing the streaming data and Hadoop File System for storage. The real-time based root cause identification architecture is presented in figure 1.

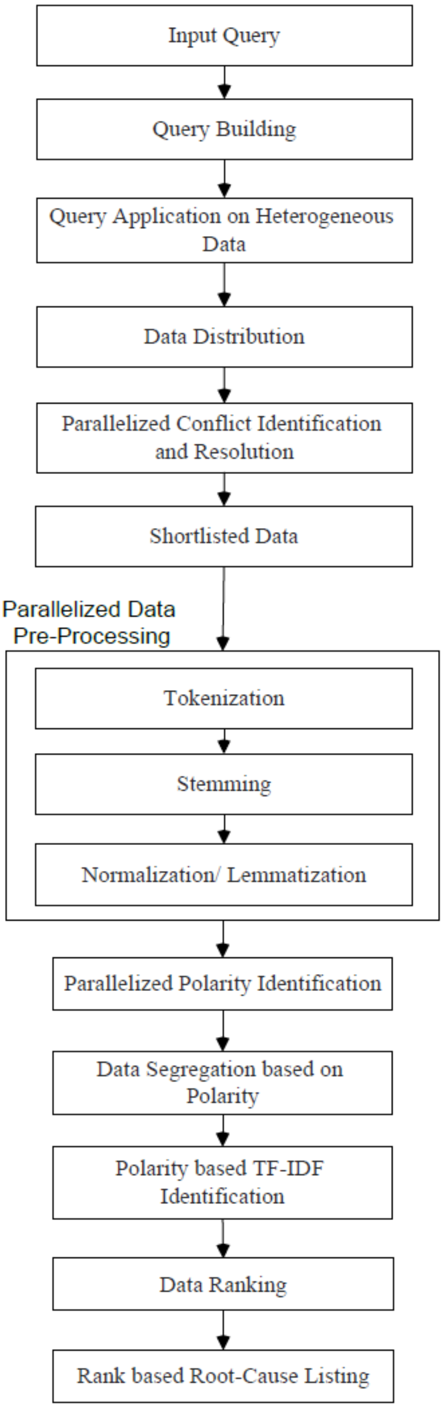
Input Query
Query Building
Query Application on Heterogeneous Data
Data Distribution
Parallelized Conflict Identification and Resolution
Shortlisted Data
Parallelized Data Pre-Processing
Tokenization
Stemming
Normalization/ Lemmatization
Parallelized Polarity Identification
Data Segregation based on Polarity
Polarity based TF-IDF Identification
Data Ranking
Rank based Root-Cause Listing

Fig -1: Root Cause Analysis - Architecture

Users are required to provide the keywords as the input query. The input query is passed to the query builder. The query builder embeds the query into corresponding API calls, SQL/LINQ queries etc. The queries corresponding to each of the data sources is applied. Every data source delivers results in its corresponding format, along with the metadata. The first phase of the proposed architecture is to identify the appropriate content for processing. This is performed by using the metadata. The contents are analyzed in a rough manner and conflicts, if any are identified and resolved. Previous paper [18] of the authors covers this phase in detail. Conflicts are identified and eliminated and the shortlisted data is passed to the next phase.

In-order to identify root causes, data needs to be examined in context level. Data obtained from the data sources are in varied formats and are inflected with metadata and several other components. Hence it becomes mandatory to convert all the data to a unified format and eliminate the unnecessary components from the data. This is performed by the parallelized data pre-processing module.

3.1 Parallelized Data Pre-Processing

Data pre-processing is the process of analyzing the components of the data to retain the appropriate components and to eliminate the others. This process plays a vital role in cleaning the input data, making it appropriate for the actual process to be applied upon it. The pre-processing phase has been parallelized and is implemented in the Spark architecture to improve its processing capabilities. The pre-processing components include tokenization, stemming and normalization/ lemmatization.

3.1.1 Tokenization

Tokenization is the process of identifying the individual components of a text for detailed analysis. This module works by dividing the text on the basis of several acquired heuristics. The most prominent division heuristics include splitting on the basis of space, commas and full stop. A set of contiguous and meaningful tokens are generated by this module.

3.1.2 Stemming

The generated tokens are to be further filtered and reduced to their basic forms, in-order to identify their polarity levels. This is carried out in the stemming module. Though the tokenization module divides and provides a set of meaningful tokens, not all the tokens are meaningful, when analyzed in isolation. English language is used as the base analysis language. According to the grammar rules, constructing a sentence in English requires prepositions and conjunctions. However, these components do not possess any meaning without the major components. Such words needs to be eliminated so as to reduce the computational complexity of the root-cause analysis process. Such words are termed as stop-words. Stop-words are eliminated and the other components are analyzed by the stemming module.

Stemming is the process of reducing a word to its base form, also called the seed word. Words are usually inflected with prefixes and suffixes in-order to fit them into sentences. A single word can constitute several such variants. Hence it becomes mandatory to reduce the word to its base form before analysis. Stemming is basically a pattern matching process that uses regular expressions for identifying the patterns and eliminating them. The earliest stemming technique was proposed by Lovins in [19]. The next widely used stemming technique was proposed by Martin Porter in [20]. The second variant [21] of this technique, Porter 2 is still one of the widely used stemming techniques. This work uses a modified form of the Porter stemmer, parallelizes it and the stemming process is carried out.

3.1.3 Normalization/ Lemmatization

As stemming is carried out using regular expressions, they tend to eliminate parts of some proper words along with the inflected words. This tends to make some words meaningless. This warrants the need for normalization. Normalization also deals with correcting the inflected words that have been modified due to the colloquial forms of expressions. Normalization uses a reference repository to identify words and replaces them appropriately, making all the ripped off words and the inflected words meaningful.

3.2 Parallelized Polarity Identification

Identifying polarity is the major functionality of the proposed work. Polarity identification is to be performed in the entire text as a whole. This is performed by identifying the polarity levels of each of the component of the text and then by aggregating the entire polarity values to obtain the final polarity of the text. Polarity identification is performed by using the SentiWordNet polarity repository [22]. The SentiWordNet is a human annotated data repository, hence it has very high reliability levels. Though polarity is expressed in terms of its magnitude (positive or negative), every term has a level of positivity and negativity associated with it. Hence polarity of a data is identified by incorporating their positive and negative levels and performing aggregations accordingly.

$$Polarity_d = \sum_{i=1}^{n} \left(Polarity_{(pos,i)} - Polarity_{(neg,i)} \right)$$

Where n is the number of tokens in the document $Polarity_{(pos,i)}$refers to the positive polarity associated with the term i and $Polarity_{(neg,i)}$ refers to the negative polarity associated with the term i. The entire process of polarity identification is parallelized, making it faster than the conventional sequential algorithms. Data parallelism is the current requirement, as huge amount of data is involved in the process and the next level process can be completed only after completing the current phase. Spark operates on the basis of data parallelism, making the polarity prediction efficient.

3.3 Polarity based Root Cause Identification

The major reason for predicting polarity of a process is that it identifying the root causes alone is not sufficient for the next level analysis. The root cause must also contain information about whether it depicts the cause from positive magnitude or negative magnitude. This plays a major role in the decision making process. Hence the data is segregated based on their polarity magnitude. The level of polarity is in-appropriate in this context. Neutral results (exhibiting zero polarity) are also incorporated into the positive data.

Operations are carried out in-parallel for the positive and negative sets from this phase onwards. Following the segregation process, is the significant term identification phase. Even though stop-words are eliminated in the stemming phase, not all the terms correspond to high significance in the text. In-order to identify the root cause, it is mandatory to list terms in the order of their importance levels.

Term Frequency (TF) and the Inverted Document Frequency (IDF) [23] levels are used to identify the significant terms in a text. TF-IDF is calculated using eq 2.

$$tfidf(t,d,D) = tf(t,d) \times idf\ (t,D) \qquad (2)$$

Term Frequency (TF) and the Inverted Document Frequency (IDF) are calculated using (3) and (4)

$$tf(t,d) = \frac{f\,(t,d)}{count\ (w,d)} \qquad (3)$$

where, *f(t,d)* refers to the number of times the word *t* is containedin the document *d* and *count(w,d)* refers to the total number of words contained in the document *d*.

$$idf(t,D) = log\frac{N}{|\{d \in D: t \in d\}|} \qquad (4)$$

where, N is the total number of documents in the corpus, and $|\{d \in D: t \in d\}|$ is the number of documents that contains word *t*. If the term is not in the corpus, then it will lead to a divide-by-zero error, hence it is also common to adjust the denominator to1+$|\{d \in D: t \in d\}|$.

Ranking of terms is performed using the rank values obtained from eq 2. A threshold limit is set and the terms with TF-IDF values greater than the specified threshold are passed as the final results.

4. RESULTS AND DISCUSSION

Experiments were conducted with data retrieved from Google API [24] and New York Times API [25]. The user's query is converted according to the API's requirements and are applied. Data and metadata retrieved from the API for the given query are collected and conflict identification and eliminated are performed. This process is followed by data preprocessing and then root-cause identification.

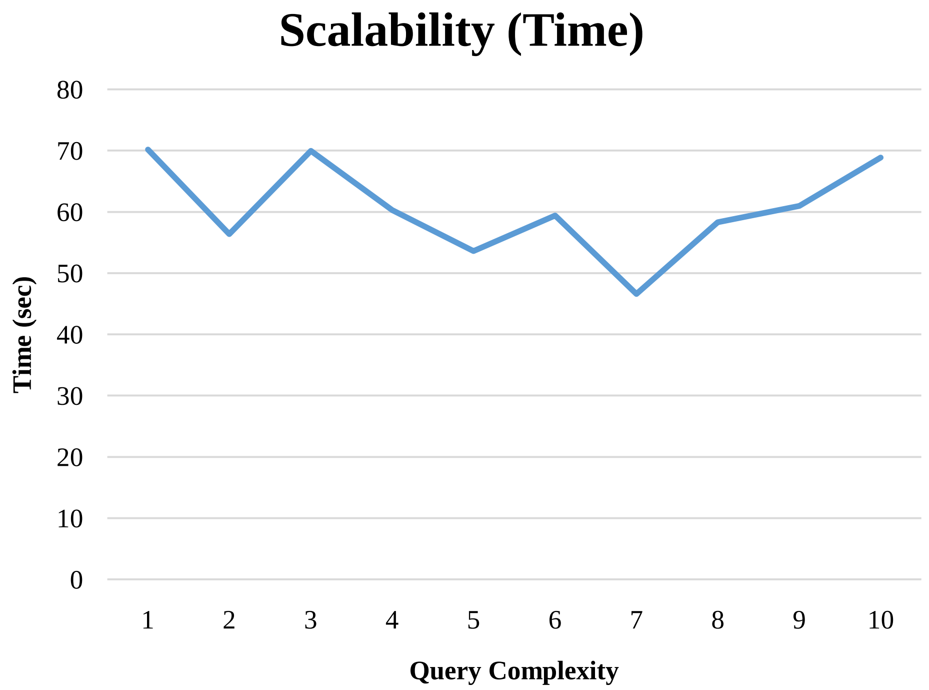

Fig -2: Scalability (Time)

Scalability was discussed as one of the major issues, due to the unpredictable nature of the social networks. The velocity and the volume of data remains unpredictable. Hence the proposed architecture was analyzed with queries of varying complexities, ranging from 1 to 10. Complexity of the query is identified by analyzing the number and the type of results returned by them. Processing time required by the entire architecture is measured as in figure 2. It could be observed from the figure that irrespective of the complexity of the query, the time of processing remains at an average level of 60 seconds. Slight fluctuations are attributed to the network delays. Such delays are very common in networks and it is mandatory to consider them, hence the average time of 60 seconds with an error level of 10% is assumed as the time complexity.

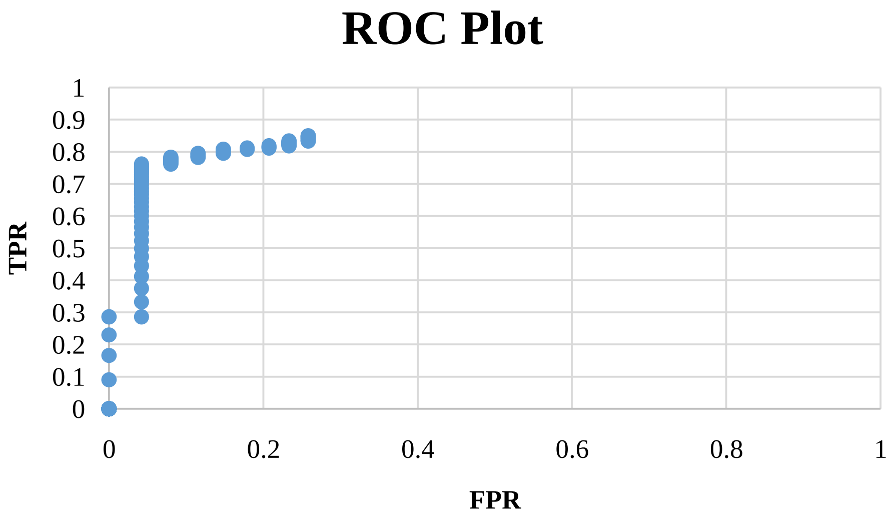

Fig -3: ROC Plot

ROC and PR plots are used to identify the efficiency of the sentiment prediction module. ROC plot (figure 3) represents the ratio between the true positive levels and the false positive levels. The requirement of a good algorithm is to exhibit high true positive levels and low false positive levels. It could be observed that the true positive levels of the proposed architecture increases to a maximum level of 0.86, while the false positive levels remains at a low level of < 0.3, exhibiting the high efficiency of the sentiment prediction module.

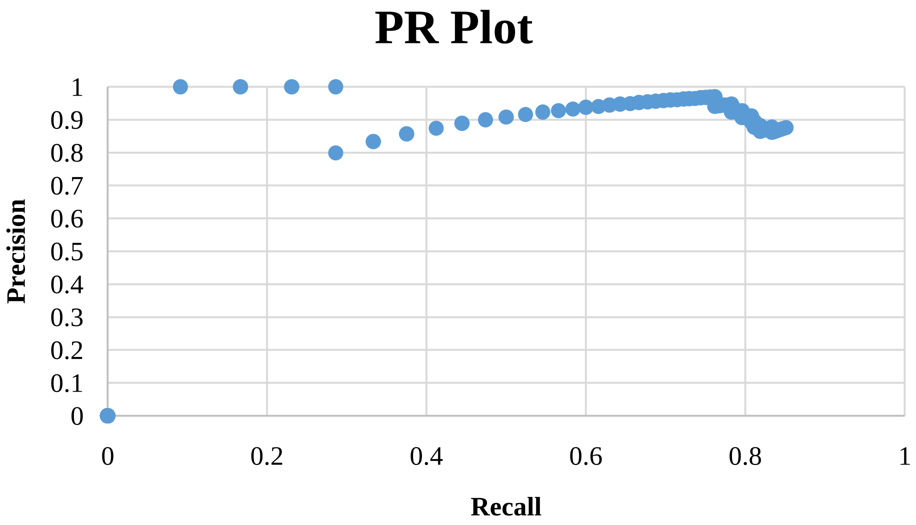

Fig -4: PR Plot

PR plot exhibits a ratio between the precision and recall levels (figure 4). The requirement of a good algorithm is to exhibit high precision and recall levels. It could be observed that the precision levels of the proposed architecture reaches a maximum of 1, representing 100% accurate retrieval levels and the recall levels reach a maximum of 0.86. Several other metrics used for analysis are tabulated and presented in table 1. It could be observed that the proposed technique exhibits an accuracy of 82% and an F-Measure of 0.86, exhibiting the highly efficient nature of the architecture. However, low true negative rates indicate scope for improvements in the architecture.

Table 1: Analysis Metrics

Metric	Value
Accuracy	0.816327
F-Measure	0.863636
TNR	0.741935
FNR	0.149254

5. CONCLUSION

Root cause analysis plays a major role in business decision making. However, this process is made complex due to the voluminous nature of the base data. This paper presents a parallelized in-memory processing technique that identifies a list of the major root causes for the proposed query and also provides a segregated view of the root causes on the basis of their polarity. The proposed architecture uses Spark, a parallel in-memory processing architecture in-order to provide real time results on streaming data. The experimental results exhibit high scalability levels and high prediction levels. However, the true negative levels remain low.

Future works will be based upon improving the true negative rates. On analysis, it was identified that the low true negative rates are attributed to sarcasm in text. Hence further works will also concentrate on identifying and rating results on the basis of context as well as sarcasm.

REFERENCES

[1] R. Piryani, D. Madhavi, & V. K. Singh, “Analytical mapping of opinion mining and sentiment analysis research during 2000–2015.” *Information Processing & Management*, 53(1), 122-150, 2017.

[2] L. Yidong, "Opinion Search and Opinion Mining: A New Type of Information Technology Bringing about New Scientific and Talent Revolution." *In Instrumentation, Measurement, Circuits and Systems*, pp. 1019-1024. Springer Berlin Heidelberg, 2012.

[3] A.Golande, R. Kamble, and S. Waghere, “An Overview of Feature Based Opinion Mining.” In *The International Symposium on Intelligent Systems Technologies and Applications* (pp. 633-645). Springer International Publishing, 2016.

[4] Y. Liu, J.W. Bi, & Z. P. Fan, “Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory.” *Information Fusion*, *36*, 149-161, 2017.

[5] M. Khorsandi, C. Skouras, K. Beatson, and A. Alijani, “Quality review of an adverse incident reporting system and root cause analysis of serious adverse surgical incidents in a teaching hospital of Scotland.” *Patient safety in surgery*, *6*(1), p.1, 2012.

[6] K. Guo, Y. Sun, and X. Qian, “Can investor sentiment be used to predict the stock price? Dynamic analysis based on China stock market.” *Physica A: Statistical Mechanics and its Applications*, *469*, pp.390-396, 2017.

[7] D. Aldoğan, and Y. Yaslan, “A comparison study on active learning integrated ensemble approaches in sentiment analysis.” *Computers & Electrical Engineering*, 2016.

[8] S. Rathi, S. Shekhar, and D.K. Sharma, “Opinion Mining Classification Based on Extension of Opinion Mining Phrases.” In *Proceedings of International Conference on ICT for Sustainable Development*(pp. 717-724). Springer Singapore, 2016.

[9] S. Takahashi, A. Sugiyama, and Y. Kohda, “A Method for Opinion Mining of Coffee Service Quality and Customer Value by Mining Twitter.” In *Knowledge, Information and Creativity Support Systems* (pp. 521-528). Springer International Publishing, 2016.

[10] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K.C. Chatzisavvas, “Sentiment analysis leveraging emotions and word embeddings.” *Expert Systems with Applications*, *69*, pp.214-224, 2017.

[11] T. Mikolov, K. Chen, G. Corrado, & J. Dean, “Efficient estimation of word representations in vector space.” CoRR, abs/1301.3781, 2013.

[12] J. Pennington, R. Socher, & C. D. Manning, “Glove: Global vectors for word representation.” In EMNLP (pp. 1532–1543), 2014.

[13] X. Li, H. Wang, and X. Yan, “Accurate recommendation based on opinion mining.” In *Genetic and Evolutionary Computing* (pp. 399-408). Springer International Publishing, 2015.

[14] J. Zhou, T. Luo, “Towards an Introduction to Collaborative Filtering.” In: Computational Science and Engineering, pp. 576–581. IEEE, Los Alamtios 2009.

[15] L. Si, R. Jin, “Unified Filtering by Combining Collaborative Filtering and Contend-Based Filtering via Mixture Model and Exponential Model.” In: Knowledge Manage, pp. 156–157. ACM, New York 2004.

[16] D. Jannach, M. Zanker, A. Felfering, “Recommender Systems: An Introduction.” Cambridge University Press 2011.

[17] D. Vilares, C. Gómez-Rodríguez, and M.A. Alonso, 2016. “Universal, unsupervised (rule-based), uncovered sentiment analysis.” *Knowledge-Based Systems*, 2016.

[18] C. S. Britto, and S. P. Victor, “Fast and Efficient Conflict Identification and Resolution in Huge Streaming Data.” International Journal of Computer Applications 146(1):10-15, July 2016.

[19] J.B. Lovins, Development of a stemming algorithm (p. 65). Cambridge: MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[20] M. F. Porter, “An algorithm for suffix stripping.” Program, 14(3), pp.130-137, 1980.

[21] https://tartarus.org/martin/PorterStemmer/

[22] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.” In LREC (Vol. 10, pp. 2200-2204), 2010.

[23] R. Baeza-Yates, and B. Ribeiro-Neto, Modern information retrieval (Vol. 463). New York: ACM press, 1999.

[24] https://developers.google.com/apis-explorer/#p

[25] http://developer.nytimes.com/docs/read/article_search_api_v2

BIOGRAPHIES

S. Charles Britto received MCA degree from St Joseph's College, Trichy, received M. Tech(QAT) from NIIT Trichy and received M. Tech(CS) from Bharathidasan University, Trichy. He is currently an Associate Professor of Computer Science Department of Nehru Memorial College, Puthanampatti, Trichy.

Dr. S. P. Victor, Associate Professor in Computer Science and Dean of Sceience,St. Xavier's college (Autonomous), Palayamkottai, Tirunelveli, received his M.C.A. degree from Bharathidasan University, Tiruchirappalli, and M.E. (CSE) degree from Anna University, Chennai. He received Ph.D. degree in Computer Applications from M. S. University, Tirunelveli, in 2005. He is teaching Computer Science since 1988. He has been the Head of the Department of Computer Science and the Director of the Computer Science Research Centre for six years. The M.S. University, Tirunelveli has recognized him as a research guide. So far 14 candidates have completed their Ph.D. degree under his guidance, and 8 candidates are pursuing research. He has published more than 80 research papers in international journals. He has organized Conferences and Seminars at the state and national levels.