Term Categorization Using Latent Semantic Analysis for Intelligent Query Processing

K. Selvi, P. Shobharani, M. L. Aishwarya, M. Rajkuumar

Abstract: ***With the rapid improvement in the field of social networks, a huge amount of small size texts are generated within a fraction of a second. Understanding and categorizing these texts for effective query processing is considered as one of the vital defy in the field of Natural Language Processing. The objective is to retrieve only relevant documents by categorizing the short texts. In the proposed method, terms are categorized by means of Latent Semantic Analysis (LSA). Our novel method focuses on applying the semantic enrichment for term categorization with the target of augmenting the unstructured data items for achieving faster and intelligent query processing in the big data environment. Therefore, retrieval of documents can be made effective with the flexibility of query term mapping.***

Keywords : Machine Learning; Natural Language Processing; Knowledge Engineering; Term Similarity; Latent Semantic Analysis; Text Categorization; Query Processing.

I. INTRODUCTION

This is an International reputed journal that published research articles globally. All accepted papers should be formatted as per Journal Template. Be sure that Each author

Semantic network is a network which signifies the semantic relation between concepts. Semantic relationship between terms is being used to search and retrieve web documents within the network. The terms are said to be conceptually similar if they are close in meaning. These terms can be either single words or multi-words. For instance, “Ford” and “Mercedes” are similar as they belong to car companies whereas “Trekking “and “Mountains” are closely related but not similar [7]. Search engines play a vital role in semantic network in order to retrieve the relevant documents based on the query of the user. Social networks services (Facebook, Twitter) and online research communities are being benefited with the advancement of semantic web. Semantic enhancement is the process of associating the semantic tag information in terms of relationships, concepts, properties, and events in ontology.

With the explosion of these services and other communities, millions of short texts are generated within a fraction of a second. These texts include short messages and comments. As they are generated from web applications, it may contain information that are not useful and does not have any value. To compute similarity between these texts is quiet difficult. The major challenging task is to understand and categorize these short texts. Several reviews and surveys concluded that the existing taxonomies require knowledge which is to be well defined so that classification can be performed [3].

The semantic information extraction plays an essential role in pinpointing the contextual information of the large-scale databases, which facilitates and enables the analysis of the unstructured data. Our contribution includes

- Latent Semantic Analysis which captures only true relation between the documents using Singular Value Decomposition (SVD) by computing the similarity among the terms
- This approach provides flexibility as terms and documents are mapped onto the same K-dimensional space and reduce feature distribution and noise using SVD.
- This proposed system also calculates similarity between terms for better text categorization.

Natural language query processing is the most challenging processes. Hence an effective query processing is essential to retrieve the results from the large-scale databases. Big data analytics and query processing techniques require that the users have the knowledge about underlying data schema. The techniques lack in exploring high semantic expressiveness of core and domain ontology and identifying the hidden relationship in the big data. Also, the query processing methods depend on the inference engine to increase the query performance, which is inconvenient for structured query language. Hence, the proposed system focuses on both the top level and the domain ontology with rich set of topics and its associated conceptual relations. Moreover, it targets for effectively processing the complex queries in a cost effective manner by formulating a natural language user query into the structured Resource Description Framework. It

Revised Manuscript Received on November 12, 2019.

***** Correspondence Author

K. Selvi, Assistant Professor in the Computer Science and Engineering R.M.K. Engineering College, Tamil Nadu, India.
Email: ssi.cse@rmkec.ac.in

P.Shobharani, Associate Professor in Computer Science and Engineering of R.M.D. Engineering College, Chennai, Tamil Nadu, India.
Email: psr.cse@rmd.ac.in

M. L. Aishwarya, Assistant Professor in the Department of Information Technology, R.M.K. Engineering College, Tamil Nadu, India.

M. Rajkuumar, Associate Professor in Computer Science and Engineering of R.M.D. Engineering College, Chennai, Tamil Nadu, India.
Email: mrk.cse@rmd.ac.in

enables the user to produce their queries in a smart way. Our novel approach focuses on applying the semantic enrichment on the big data sources with the target of augmenting the unstructured data items and achieving faster and intelligent query processing in the big data environment

II. RELATED WORK

Several researches are being done in the field of semantic network to compute similarity. Search engines play a imperative role in retrieval of relevant documents based on the query. To measure the conceptual relationship among query terms in is-A relationship is based on the information content proposed by Resnik [10].

An important movement in Text Mining is the grouping of texts, i.e. texts of the same or similar content in a group, i.e. to discern texts of different content. Yaxiong Li et al., 2010[15], proposed a model named as domain ontology and builds a very new conceptual space vector model in the preprocessing stage of Text Clustering by replacing the original matrix in the latent semantic analysis with concept - text matrix. In this clustering technique, document representation takes into description the semantic similarity, which partially overcome the difficulties in accuracy and effectively evaluates the degree of similarity between the text based on the frequency of words or phrases in the text.

A system for competent unsupervised example for population count proposed by Theerayut Thongkrau et al., 2010[12], the new instances are classified according to their equivalent lexical ontology concept. In relation to the processing time, it must not use many of the concepts in the lexical ontology to search. It needs to manage unlimited number of ontological concepts in each domain. This system employs a latent semantic analysis together with the vote on the appropriate term of the instance to be found. Karthick et al., 2014 [4], analyzed the effect of including multi-word frequency on the hierarchical clustering of web documents to determine its performance. It also analyze the outcome of combining link and content based representation of web documents and their interrelationship.

Traditional SVM algorithm classifies tested samples in error in the neighborhood of the optimal hyper surface. Therefore, a Hybrid based KSVM algorithm is proposed by Zhang Su-zhi et al., 2011[17], for categorization by computing the distance of the sample to the optimal hyper surface f the SVM in the feature space, and selects different algorithms for different distances. Categorization of short text is done at last.

A frequent term based text clustering approach using novel similarity measure is presented by Reddy et al., 2014[9], introduced a measure to determine the common features between two text files which is used as a similarity measure.

Qiuxing Chen et al., 2016 [8], proposed an enhanced short text classification method based on Latent Dirichlet Allocation topic model and K-Nearest Neighbor algorithm. The generated probabilistic topics help them make the texts more semantic-focused to reduce the sparseness.

To calculate the similarity between terms based on synsets, a new method is proposed by Mamta Kathuria et al., 2016[5], in which synsets are derived using online resources. The advantage of the proposed work is that, the similarity measure between terms is calculated that helps in query suggestion or replacement of one query with the most appropriate query.

Narayanan et al., 2013[6], has introduced EDA (Enhanced Distributed Algorithm) for document clustering using other similarity measures such as cosine similarity, Jaccard and Pearson coefficient. The performance of EDA is evaluated using similar performance factors to determine the accurateness and clustering quality. With the help of Distributional Term Representations (DTRs) by Cabrera et al., 2013[2], categorization is performed for short texts. It represents term based on contextual data, as statistics on document occurrence and term co-occurrences provided.

Wang et al., 2012[14], presented a Framework for improving Web search experience through the use of a probabilistic knowledge base. This framework categorizes web queries in various patterns according to the concepts as well as entities in addition to the keywords present in these queries. Further the queries are answered by the interpretation of the questions with the help of the knowledge base. Selvi K et al [20] propose an pragmatic method to find semantic similarity by means of artificial neural network by combining the different lexical patterns that extracts the semantic relation between two words accurately.

To compute semantic similarity between terms based on synsets a new method is proposed by Mamta Kathuria et al., 2016[5], in which synsets are derived using online resources. The advantage of the proposed work is that, the semantic similarity between terms is calculated that helps in query suggestion or replacement of one query with the most appropriate query.

Multi-Intelligence Data Integration Services (MIDIS) is a data integration approach that exploits Domain ontology's to annotate the big data. The Domain Ontology's contains a group of concepts connected to a specific domain; therefore, it does not effectively represent the semantic relation in the big data from heterogeneous sources [22].

The integration of the linked data leads the feasible large data analysis in supply chain management. It presents a fascinating potential external source for integrating big data using Linked data, but they mainly concentrate on the logistics domain for supply chain management [23]. An integrated method [24] combines structured, semi-structured, and unstructured data using the semantic wiki. The semantic wiki maintains rich graph and stores the metadata as RDF triples based on the combination of the data in a relational database. Linking and Processing Tool for Unstructured and Structured information (LIPTUS) [25] apply the keyword based search on associating the unstructured data with the structured data in the banking environment. To obtain the business intelligence [26], the system consolidates the ordered and unordered data based on the Online Analytical Processing (OLAP).

II. METHODS AND MATERIALS

A. Context Similarity

We use cosine similarity function F() to check whether two contexts are similar where Tt1 and Tt2 are two contexts.

$$sim(Tt1\ ,\ Tt2) = F\ (Tt1\ ,\ Tt2) \qquad (1)$$

B. Type checking

For a given pair of terms, it is necessary to identify the type and then further processing is done. As we use LSA based SVD, it captures only the accurate relationship among the documents as well as the terms in the semantic network so it is easy to identify the type.

In [7] the approach uses is-A relationship to compute the similarity between terms. For the given pair of terms, first perform type checking so that the terms are categorized as concept, entity and concept-entity. These are then converted into respective vectors based on conditional probability in is-A semantic network. But it does not specify how to categorize short texts.

Algorithm 1: Similarity Score

Input: (t1, t2): pair of terms;
ΓisA: the semantic network of isA relationship;
Γssyn: the synset data set
in $\Gamma$is A; maxD:
maximum iteration depth;
Output: a similarity score
(*t*1, *t*2);

```
if $t_1$ and $t_2$ belong to the same synset by $\Gamma_{ssyn}$ then
    Let $sim(t_1, t_2) \leftarrow 1$ and return $sim(t_1, t_2)$;
end if
Judge the type for each term;
if $t_1,t_2$ is a concept pair then
    Generate the entity vector $I_c^{ti}$ $(i \in \{1, 2\})$ of $t_i$ as defined in (2) using $\Gamma_{isA}$;
    return $sim(I_c^{t1}, I_c^{t2})$ as defined in (4);
end if
if $t_1,t_2$ is an entity pair then
    Generate the concept vector $I_e^{ti}$ $(i \in \{1, 2\})$ of $t_i$ as defined in (3) using $\Gamma_{isA}$;
    return $sim(I_e^{t1}, I_e^{t2})$ as defined in (1);
end if
if $t_1,t_2$ is a concept-entity pair then
    Collect topK concepts of the entity term $t_i$ from $\Gamma_{isA}$ as the context $C_{ti}$ $(i \in \{1, 2\})$;
    For each $c_x$ in $C_{ti}$ $(c_x \neq t_j, i \neq j, 1 \leq x \leq topk)$ do
        $sim_{cx} \leftarrow$ get the similarity between $c_x$ and $t_j$ byrepeating this algorithm iteratively if the current iteration depth is no more than maxD;
    end for
    return $\max_{cx \in Cti}\{sim_{cx}\}$;
end if
```

Short text categorization is performed using the similarity score. Algorithm 1 shows that a pair of terms (t1, t2) from synset data set is given as input. If the terms belong to the same synset, then the similarity is equal to 1. Type checking is done to find the type for each term. If the pair of terms given is a concept pair, then entity vector for the terms are generated and similarity between these vectors is found using equation(2).

$$I_c = (w1' , \ldots wk') \qquad (2)$$

If the pair of terms given is an entity pair, then concept vector for the terms is generated and similarity between these vectors is found using the below equation.

$$I_e = (w1, \ldots wk) \qquad (3)$$

If the pair of terms given is a concept-entity pair, then top-K concepts of the entity term are collected from isA relationship as Contexts. For each context, similarity measure between the context and term is calculated until current iteration depth is not greater than Max- D. Finally the maximum similarity among them is returned.

IV. PROPOSED METHOD

From the short texts, meaningful terms are captured and applied in the proposed algorithm ignoring the stop words. In general, stop words are the most common occurring words in a language. Therefore, the inputs to the algorithm are considered as term pairs to compute similarity.

In this proposed system, the usage of is-A relationship is compensated by SVD performed on the computed weights. It also overcomes the limitation of [7]. Type checking is done to find whether the given term is a concept or entity or concept-entity pair.

For each type, LSA analyzes the association between set of documents and terms. Based on the frequency of the occurrence of the terms, a matrix is constructed. To reduce the complexity of the matrix generated and make it a non-zero matrix, SVD is applied. Vectors of the terms are computed by taking any two rows of the matrix. To compute weight $a_{ij,}$ we use local and global weights of the term in the document.

$$a_{ij}=LW_{ij}\times GW_{ij}$$

where LW_{ij} is the local weight of the term i in the document j (i.e.) how many times each term appears and GW_{ij} is the global weight of the term I in the document j (i.e)how many times each term appears in entire dataset. LSA makes use of SVD of $A_{mn=}$ $[a_{ij}]_{mxn}$ matrix to capture only meaningful relationship between the documents and it is the product of three matrices.

$$A=\sum_{i=1}^{r} u_i\sigma_i v_i= [u_{1\ldots}u_r]\begin{bmatrix}\sigma & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma\end{bmatrix}$$

where u and v are the matrices of the left and right singular vector and $\sigma$is the diagonal matrix of the singular values. According to SVD, reduced space form $U_k\ {}_k\ V_k^T$is similar to the original vectors. Then their similarity can be calculated using (1). It is given by,

$$A =U_r \sum{}_r V_r^T \quad U_k\sum{}_k V_k^T$$

Algorithm 2 : Proposed Algorithm

Input: (t1, t2) a pair of terms;
Kcluster: cluster of all concepts using K-means maxD: the maximum iteration depth;
Output: a similarity score;

```
Perform type checking for each term;
if (t1,t2) is a concept pair then
  Generate concept vector for the terms using LSA;
    return similarity of concepts using(1)
end if
if (t1,t2) is an entity pair then
    Generate entity vector for the terms using LSA
    e1<-returns similarity using (1)
    Find clusters of context for the terms from Kcluster;
    e2<-returns similarity using(1)
    return max(e1, e2);
end if
if (t1,t2) is a concept-entity pair then
    Collect top K concepts of the entity term as contexts and find clusters from Kcluster;
for each concept cx in topK(cx≠tj,i≠j,1≤ x≤topK) do
  sim (cx)<-perform similarity between cx and tj by iterating the algorithm until maxD;
end for
return max(sim (cx));
end if
```

parameter max D is specified for iterating the algorithm. Its value ranges from 1 to 20. The algorithm stops iterating when the current iteration depth is not greater than maxD. At the end of this approach, similarity scores are obtained for each term and then clustered using K means algorithm. With this approach, search engines can effectively retrieve documents based on the terms.

Word1/ television/radio television/radio media / radio Word2

V. RESULTS AND DISCUSSION

Results from our experiment indicate that similarity measures can be utilized to determine the parameters for clustering and classifications. Our method has been tested in few popular search engines and found giving better results as given below in Figure 5.1.

K (“Biocon CEO”,“KiranMuzmdar Shaw”) =0.826
K (“Apple CEO”,“Tim Cook”) =0.855
K (“OracleFounder”,“Larry Ellison”) =0.781
K (“Yahoo Founder”,“Jerry Yang” =0.695
K(“FacebookFounder”,“Mark Zuckmber”)=0.099

Figure 5.1 Test Results from five search engines

Also, the function is more effective in giving low scores to pairs, which vary semantically.At the end of the training process, the matrices store final weight values for mapping inputs and outputs. During the testing process, final weights are used for processing with the vectors corresponding to word pair. In another method, final weights are obtained from the pattern itself without any initialization of the weight matrices.

It is aimed to improve the performance of text document clustering on application of Linear Regression Algorithm, which reduces the error rate by 0.8 %. K-Means Clustering algorithm is used for grouping similar documents.The data used in our experiment is WordSim353. Evaluation on training data set generated for 2 iterations is shown below in Figure 5.2

Several studies indicate that the human scores consistently have very high correlations reaching 0.9632and syntactic context perform best on WordSim353. Note that the confidence intervals are quite large in WordSim353 and few of the pair wise differences are statistically

$$A = U_r \sum_r V_r^T \approx U_k \sum_k V_k)^T \quad \textbf{(5)}$$

The features for the word pair was defined from the word co- occurrence measures. The outputs are well interpreted to identify whether the documents retrieved or clustered are relevant to the words or not. Experimental output showed that our proposed technique outperforms the human ratings with limited number of Clusters and the time taken to build, was only 0.08 seconds at the testing stage as shown in Table 5.3

significant.

```
Model and evaluation on training set
=====================================
 K Means
 Number of iterations: 2
 Within cluster sum of squared errors:
 23.150937685414707
 Missing values globally replaced with mean/mode
 Cluster centroids:     Cluster#
Attribute          Full Data          0            1
                     (24)           (16)         (8)
      HR             6.7208         6.0963        7.97
      PM             6.9788         6.3481        8.24
```

Figure 5.2 Evaluation on training data

The proposed method can be applied for any context where taxonomies are not required for query processing of text. Table 5.1 summarizes the error rates for 353 classified instances for the given data. The genetic algorithm, linear regression yields an average value of 0.58 approximately with the absolute error rate of 40 %.

```
    Average Target Value     : 0.5758305038549534
    Inverted Covariance Matrix:
    Lowest Value    = -0.34971644612476366
    Highest Value   = 0.6502835538752364
    Inverted Covariance Matrix *
     Target-value Vector: Lowest Value   =
     -0.2588713208324032
    Highest Value   = 0.27371229405845615
    Time taken to build model: 0.66 seconds
    Scheme: weka.Classifiers.functions.LinearRegression
    Correlation co-efficient            0.9632
    Mean absolute error                 0.6996
    Root mean squared error             0.87
    Relative absolute error
    39.0574% Root relative squared
    error         40.0484%
Coverage   of   cases   (0.95 level) 100%
    Total Number of Instances           353
```

Table 5.1 Error rates with Correlation Co-efficient

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

50% percentage split on supplied training set classifies into Cluster 0 and Cluster 1 as given in Table 5.2 Clusters generated

```
Class attribute:  Word1  & Word2

Classes to Clusters:                          0 1  <-- assigned to
cluster

 1 0 | television  radio                     0 1 | tennis racket

 1 0 | Arafat peace                          0 1 | media radio

 0 1 | Arafat terror                         1 0 | drug abuse

 1 0 | Arafat Jackson                        1 0 | bread butter

 0 1 | law lawyer                          1 0 | cucumber potato

 0 1 |  admission ticket                     1 0 | doctor nurse

 1 0 | shower thunderstorm                 1 0 | professor doctor

 1 0 | shower flood                       1 0 | student professor

 0 1 | weather forecast                      0 1 | football soccer

 1 0 | disaster area                    1 0 | football basketball

 1 0 | governor office                       1 0 | football tennis

 1 0 | architecture century                0 1 |
ticket
```

Table 5.2 Clusters generated

```
Test mode:    10-fold cross-validation

SimpleLogistic:

Class 0 : 1.68 +  [a2] * -0.28 + [a3] * -0.53 +[a5] *
-1.73 + [a8] *
0.7

Class 1 : 1.68 +  [a2] * 0.28 + [a3] * 0.53 +[a5] * 1.73 +
[a8] * -
0.7

Time taken to build model: 0.08 seconds
```

Table 5.3 . Results obtained for Simple Logistic classification

VI. CONCLUSION AND FUTURE ENHANCEMENTS

This paper focuses on Latent Semantic Analysis for categorization of short texts. We use Latent Semantic Analysis based SVD technique which makes the classification efficient. This method performs efficient clustering of short texts and can be applied on large scale data set. Also it improves flexibility to map terms in the semantic network. In our future work we will focus on how to apply our approach in multi-label short text classification. It is clear that adding some information about the conceptual terms between the words could improve the clustering performance. In order to deal with the problem, we can integrate background knowledge into the process of clustering text documents. We can use background knowledge in document clustering by integrating an explicit conceptual account of terms found in WordNet.

Clustering could be improved by new descriptors of the Lexicon. Areas of future work include enhancements to the segmentation algorithm that could improve accuracy and execution time. System can be hardly accelerated by new fitting algorithm. This intelligent data integration method facilitates the query processing by reducing the burden of searching process rather than exploring the enormous data sources which is one of the major challenge faced by popular search engine such as Google. Also better results can be achieved by increasing the number of keywords.

REFERENCES

1. E. Agirre, M. Cuadros, G. Rigau, and A. Soroa, “Exploring knowledge bases for similarity,” in Proc. of LREC’10, pp. 373–377, 2010.
2. Cabrera, Juan Manuel and Escalante, Hugo Jair and Montes-y-mez, Manuel, “Distributional term representations for short-text categorization”, Computational Linguistics and Intelligent Text Processing, pp. 335-346, 2013.
3. Ge Song,YunmingYe, XiaolinDu, XiaohuiHuang, and ShifuBie, “Short text classification: A Survey”,Journal of Multimedia,Vol. 9, 2014.
4. S. Karthick, S.M Shalinie, A. Eswarimeena, P.Madhumitha and T.N Abhinaya, “Effect of multi-word features on the hierarchical clustering of web documents”,Recent Trends in Information Technology (ICRTIT), International Conference, pp. 1 – 6, 2014.
5. MamtaKathuria, Payal, C. K. Nagpal , “Semantic similarity between terms for query suggestion”, Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) 5th International Conference, pp.245-250, 2016.
6. N. Narayanan, J. E. Judith, and J. JayaKumari, “Enhanced distributed document clustering algorithm using different similarity measures”, Information & Communication Technologies (ICT), IEEE Conference, pp. 545-550, 2013.
7.PeipelLi, HaixunWang, KennyQ.Zhu, ZhongyuabWang, Xuegang Hu, and XindongWu,”A large probabilistic semantic network based approach to compute term similarity”,IEEE Transactions on Knowledge and Data Engineering,2015.
8. Qiuxing Chen, Lixiu Yao, Jie Yang , “Short text classification based on LDA topic model”, Audio, Language and Image Processing (ICALIP) International Conference, pp.749-753, 2016
9. G.S. Reddy, T.V. Rajinikanth and A.A. Rao, “A frequent termbased text clustering approach using novel similarity measure”, Advance Computing Conference (IACC), IEEE International, pp. 495-499, 2014.
10. P.Resnik, “Using information content to evaluate semantic similarity in a taxonomy,” in Proc. of IJCAI’95, 1995, pp. 448–453,1995.
11. Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, “Short text conceptualization using a probabilistic knowledgebase,” in Proc. Of IJCAI’11, pp. 2330–2336, 2011.
12.Theerayut Thongkrau and Pattarachai Lalitrojwong, “Classifying instances into lexical ontology concepts using latent semantic analysis”, Computer and Automation Engineering (ICCAE) 2nd International Conference, Vol.1, pp.66- 70, 2010.
13. P.Turney.”Measuring semantic similarity by latent relational analysis”, In Proc. of IJCAI’05, Pp.1136–1141, 2005.
14. Y. Wang, H. Li, H. Wang, and K. Q. Zhu, “Concept-based web search”, in ER, pp. 449–462,2012.
15. YaxiongLi, Jianqiang Zhang and Dan Hu, “ Text Clustering Based on Domain Ontology and Latent Semantic Analysis”, Asian Language Processing (IALP) International Conference , pp.219-222, 2010.
16. Yung-Shen Lin, Jung-Yi Jiang and Shie-Jue Lee, “A similarity measure for text classification and clustering”, IEEE Transactions on Knowledge and Data Engineering,Vol.26, pp. 1575-1590, 2014.
17. Zhang Su-zhi and Sun Pei-feng, “A new short text categorization algorithm based on improved KSVM”, Communication Software and Networks (ICCSN), 3rd IEEE International Conference, pp.154-157, 2011.
18.http://people.revoledu.com/kardi/tutorial/Similarity/Stringinstance.html#TextSimilarityCalculator.

19. https://en.wikipedia.org/wiki/Semantic_network
20. Selvi K, Suresh R M, " An Efficient Technique for Document Clustering Using Intelligent Similarity Measure", RJASET-2014. Pp. 2320-2328.
21. Selvi K, Suresh RM. "Document clustering using artificial neural networks", National Journal on Advances in Computing and Management. 2008; 5(1):1–5.
22.K.Selvi, R.M.Suresh (2016), "Strategies for Effective Document Clustering using Modified Neural Network Algorithm", Journal of Computational and Theoretical Nanoscience, Vol 13(7), July 2016, ISSN: 1546-1955 : EISSN: 1546-1963,American Scientific Publishers
23. K. Selvi, R.M. Suresh(2016), "Fuzzy Concept Lattice for Ontology Learning and Concept Classification" Indian Journal of Science and Technology, vol 9(28), July 2016 .ISSN: 0974-6846 , e-ISSN: 0974-5645, Indian Society for Education and Environment Publication
24. P.Shobha Rani, R .M. Suresh and R. Sethukarasi , "Multi-level Semantic Annotation and Unified Data Integration using Semantic Web Ontology in Big data Processing " Cluster Computing The Journal of Networks, Software Tools and Applications, Volumes20 , Issues77 August 2017 ISSN: 1386-7857 (Print) 1573-7543 (Online) Annexure I ,Impact Factor 2.040.
25. P.Shobha Rani, R .M. Suresh and R. Sethukarasi "Semantic Annotation of Summarized Sensor Data Stream for Effective Query Processing" The Journal of Supercomputing ,An International Journal of High-Performance Computer Design, Analysis, and Use, ISSN: 0920-8542 (Print) 1573-0484 (Online), Annexure I, Impact Factor: 1.326.
26. M.RajKumar, R.M.Suresh and R.SasiKumar "An effective cluster based data dissemination in a hybrid cellular ad hoc network", Concurrency Computat Pract Exper. 2018;e5125. wileyonlinelibrary.com/journal/cpe © 2018 John Wiley & Sons, Ltd. 1 of 11,https://doi.org/10.1002/cpe.5125
27. Dr.P.Shobha Rani, "Smart Video Surveillance" International Journal of Engineering Science and Computing (IJESC) (e ISSN: 2323-3361) Vol.9 Issue No: 3 March 2019
28. P.ShobhaRani," Protecting Respondents Identities in Microdata Release , SurajPunj Journal for MultiDisciplinary Research (SPJMR) ISSN: 2394-2886, Vol.9 Issue No: 3 March 2019

AUTHORS PROFILE

First Author S. Selvi received her B.E. (Computer Science and Engineering) from Bharathiar University , M.E. from the Anna University and PhD from Sathyabama University. She is working as Associate Professor of Computer Science and Engineering at RMK Engineering College. She has more than 10 paper publications in reputed journals whose papers has been cited in Google Scholar by authors working in the area of web content mining. Her research area includes Artificial Neural Networks, Web Mining and Soft Computing. A Life member of ISTE and also member of CSI, CSTA, IIEANF, EI etc.

Second Author Dr. Pacha Shobha Rani is working as Associate Professor in Computer Science and Engineering of R.M.D.Engineering College, Chennai, Tamil Nadu, India. She pursued Degree in Bachelors of Engineering (CSE) from Bangalore University and Masters in Computer Science and Engineering from Anna University, Chennai. Presently she is pursuing her Ph.D., in the area of Ontology oriented Big Data. She has 5 years of working experience in the industrial field. She is in the teaching profession for the past 15 years. She also completed her ORACLE Certification. Her areas of interest include Data Warehousing and Data Mining, Big Data. She has published two research papers in journals and conferences. She has guided four Master of Engineering projects.

Third Author M.L.Aishwarya was born in 1992.She received her M.E degree in Computer Science and Engineering and B.Tech degree in Information Technology from R.M.K. Engineering College, Anna University in 2016 and 2014 respectively. She is currently working as Assistant Professor in the Department of Information Technology. Her research interests include data mining and knowledge management.

Fourth Author M.RAJKUMAR is working as Associate Professor in Computer Science and Engineering of R.M.D.Engineering College, Chennai, Tamil Nadu, India. He has completed B.E. Computer Science in the year 2003 at Dr.M.G.R.Enginnering college, Chennai. He has completed M. Tech Computer Science in the year 2007 at Dr.M.G.R University, Chennai. Presently he is pursuing his Ph.D., in the area of Ad-hoc Networks. He served as lecturer in Vel's Srinivasa Engineering & Technology college, Chennai for six months from July 2003 to NOV' 2004 and He also served as a lecture in S.M.K.Fomra Institute of Technology from dec'2004 to may'2005.Now he is working as lecturer in R. M. D Engineering College, Kavaraipettai, Chennai from June 2005 to till date. He has 12 years of experience in teaching field. His area of interest is Operating systems and mobile computing. Life Time Member of ISTE.