Text Graph- An Enhanced Graph Fusion Model for Document Clustering

M.Uma Maheswari, J.G.R.Sathiaseelan

Abstract: Text clustering is a well-known method for refining the eminence in information retrieval, which groups a huge number of unordered text documents into the subgroup of associated documents. It is a contemporary test to investigate minimized and meaningful experiences from substantial accumulations of the unstructured content reports. Different clustering techniques are in use to make the clusters in the text document accessible. This paper introduces a new technique of document clustering based on graph model. The collection of documents is denoted as the graphical network in which the node represents a document and an edge represents the similarity between the two documents. This paper intends a TextGraph algorithm based on the graph structure. The unstructured documents contain a vast number of features; it must be reduced before graph construction. The count and semantic-based feature reduction methods are used to select the vital features. Based on this feature, the algorithm constructs the text graph structures. This paper combines these (word count and semantic) text graph structure to generate a fusion graph model. In, fusion model, each document is associated to its k-nearest neighbors with weighted edges. Finally, on the fused TextGraph, the clustering is performed to group the documents. Experimentations are accompanied on real-time text datasets. The outcomes demonstrated that the proposed fusion graph model overpowers the prevailing methods and improves the outcome of text document clustering techniques in rapports with the purity and normalized mutual information.

Keywords: Text document clustering, graph model, feature selection, semantic word frequency.

I. INTRODUCTION

Clustering in general is a significant and valuable technique that inevitably organizes a collection with a vast number of data objects into much smaller number of intelligible groups. Document clustering is widely utilized as a part of the zones of content mining and data recovery. Clustering especially underpins in combining the documents fundamentally to enhance the recovery and perusing those documents. The investigation of the issues in the document clustering is identified with the pertinence to the content area. Text document clustering is a determination of text documents with the specific word dominion. So each clustering of the documents called the clustering of text types of a specific word's occurrences. [1]. Clustering is unsupervised learning it means there is no need of human interference for clustering of documents.

There are many clustering techniques has been proposed, each implementing a convinced approach for sensing the group formation in the data, such as K-means algorithm[2], Expectation Maximization and hierarchical clustering[3]. They can be distributed into many categories: [4] partition (k-means, k-mediods), hierarchy (CURE, ROCK), fuzzy theory (FCM, FCS), density (DBSCAN, OPTICS), graph theory (CLICK, MST). In document clustering, the text is characterized by statistical and semantics models [5]. The delineation of documents known as bag-of-words considers each document as a way in a high-dimensional space; every component of this vector focuses to a single word (or, all the more by and large, highlights) in the document clustering. This portrayal depends on the Vector Space Model [6], where vector segments speak to certain element weights. VSM fabricates a model in view of word recurrence [7]. The semantic portrayal is a semantically arranged model. It depends on the semantic relationship among the ideas.

The old-fashioned clustering algorithm use any one of the text representation method i.e., statistical or semantics [8]. This paper combines these two text representation to form a fusion model. A network graph representation of the text pool is done by representing the text repository as a node and the edge between two nodes shows the similarity between two documents [9]. Two graphs are constructed using statistical and semantics text representation. These two graphs are combined to form a graph fusion model. Based on this fusion graph model the documents are cluster. To cluster documents, a recursive deletion of lowest similarity weighted edge [10] is done until desired numbers of clusters are found.

This paper put forward a TextGraph approach that cluster documents based on graph structure model. The division of the paper is described as follows. Section 2 enlightens the proposed TextGraph clustering and Section 3 deliberates the performance analysis of proposed results. Finally Section 4 recaps the work which has been done.

Revised Manuscript Received on May 28, 2019.

M.Uma Maheshwari, Research Scholar, Department of Computer Science, Bishop Heber College, Tiruchirappalli, TN, India.

Dr.J.G.R.Sathiaseelan, Associate Professor & Head, Department of Computer Science, Bishop Heber College, Tiruchirappalli, TN, India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

II. PROPOSED METHODOLOGY

In this section the proposed document clustering algorithm based on graph fusion model are explained. Figure 1 shows the architecture of proposed graph fusion model document clustering.

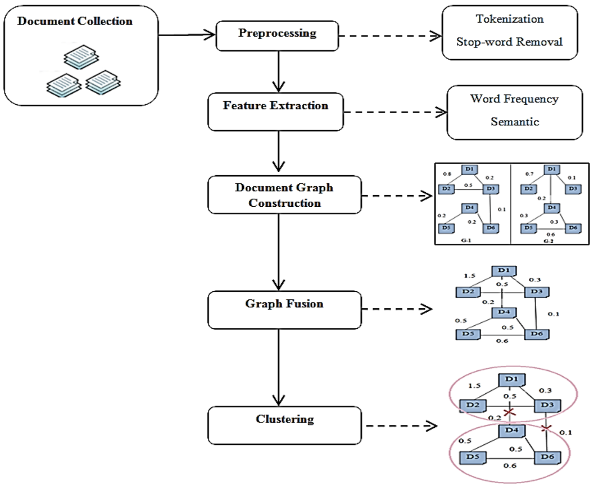

Fig. 1. System Architecture

The main workings of the system model are as in the following:

A. Text Pre-processing

Any text exploration strategy, for example, document recovery, document clustering, text component selection, et cetera, requires to change over the document substance to end up practicable arrangement. In this procedure, the gathered documents are pre-processed.

Tokenization is the advancement of cleaving an outpouring text documents into words [11] or terms and wiping out the void succession, in which each word or image is taken from the principal character to the last character, which is known as a token [12]. Stop words such as an, an, are, as, it, or are thought not to pass on any significance, are expelled from the content. They commonly take some part of the document and diminishing the number of highlights, in that route, prompting the lessened execution of the text clustering system [13].

B. Feature Extraction

In this process, the important and meaningful terms are mined from the preprocessed documents. Part-of-speech (POS) [14] classification is the method of diffusing a word to its linguistic class, so as to comprehend its character within the sentence. Traditional POS deals with the nouns, verbs, adverbs, conjunctions, etc. Initially the noun words are selected as term or word [15]. Two types of feature extraction methods are used: word frequency and semantic frequency.

```
  **Algorithm 1:** Feature Extraction
Input: Document Collection D= {$d_1$, $d_2$, $d_3$, .., $d_n$}
  Terms T= {$t_1$, $t_2$, $t_3$, …, $t_m$}
Output: Extracted Terms ET= {$et_1$, $et_2$, $et_3$, …, $et_k$}
    Word Frequency based Extraction
For each $d_i$ in D
  For each $t_j$ in T
  Compute Term Frequency tf($d_i$,$t_j$)
      EndFor
 EndFor
For each $t_j$ in T
  Compute Document Frequency df(j)
EndFor
 Compute tfidf score
```

$$tfidf(d_i,t_j)=tf(d_i,t_j)*\log\left(\frac{n}{df(t_j)}\right)$$

```
 Select maximum score terms
For each $t_j$ in T
  Add $t_j$ to queue Q
  $t_j$Fre= find $t_j$ frequency
 EndFor
 While (Q is not empty)
  t1= Q.front()
    Find the synset S for t1
Find the Hyponyms H1 for t1
    Find the Hypernyms H2 for t1
    Find the Glosses G for t1
  sum= $t_j$Fre/Σ(S, H1, H2, G)
  If sum>0
     Term t1 is add to feature list
  EndIf
  Q.pop()
 EndWhile
```

In this paper the features are extracted based on word frequency and semantic frequency. Algorithm1 explains the feature extraction algorithm. In word frequency method, the traditional Term Frequency Inverse Document Frequency (TF-IDF) technique helps in getting the score for every single term. It reflects the significance of a term to the document [16]. The minimum score value terms are removed from the term list. WordNet is used to find out the semantic frequency based features. Synset, Hyponyms, Hypernyms and Glosses are used to compute the semantic frequency of terms.[17].

C. Document Graph Construction

. Generally a graph G is defined as G= (N, E, W), Where N is set of Nodes, E is set of edges and W is set of edge weight [18]. A TextGraph is defined as TG= (D, E, SW) where D is set of documents considered as a Node, E is set of edges and SW is set of similarity weight between documents (nodes).

```
**Algorithm 2.** Document Graph Construction
Input: Documents D, Terms T
Output: Document Graph
 Construct Document Term Matrix $DTM_{d*t}$
For each document $d_i$ in D
  For each document $d_j$ in D
    For each term $t_k$ in T
      if(DTM($d_i$,$t_k$)!=0 && DTM($d_j$,$t_k$)!=0)
$W_{ij}$=Σ (DTM ($d_i$,$t_k$), DTM ($d_j$,$t_k$))
      EndIf
    EndFor
  EndFor
EndFor
 For each document $d_i$ in D
  For each document $d_j$
in D
    if($W_{ij}$>0)
  Create node $d_i$ and $d_j$
if not exists
```

```
             Add edge between $d_i$ and $d_j$ with
Weight $W_{ij}$
    EndIf
  EndFor
 EndFor
```

Algorithm 2 explains the document graph construction algorithm. The input to the algorithm is document collection and terms extracted from word frequency and semantic frequency. Initially the document term matrix was constructed (DTMd*t)[19]. In DTM, the row represents document and the column represents the terms, cell values are word and/or semantic frequency. Figure 2 shows the process of document graph construction.

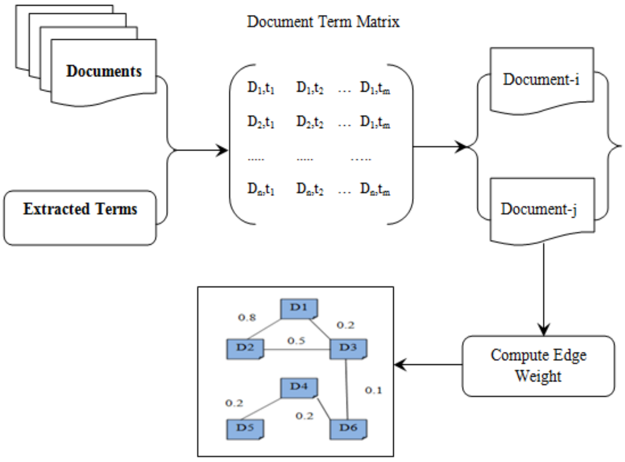

Fig. 2. The process of graph construction.

D. Graph Fusion

There are two document graphs are constructed based on word and semantic feature frequency [20]. These two graphs are combined to form a graph fusion. Algorithm 3 explains the graph fusion algorithm.

```
Algorithm 3. Graph Fusion Algorithm
Input: $G_W$= ($D_1$, $E_1$, $W_1$), $G_S$ ($D_2$, $E_2$, $W_2$)
Output: Fused Graph $G_F$= ($D_f$, $E_f$, $W_f$)
Initialize the node set $D_f$ = {}, the set of $E_f$ = {}
and the set of weights $W_f$ = {}
Search every node of $G_W$ and $G_S$, comparing these
corresponding nodes, and taking the common nodes as
the nodes G'
 While (G' has no nodes)
   Get any two nodes in G' $n_1$ and $n_2$
   Check there is an edge E' between $n_1$ and $n_2$ in $G_W$ and
Gs
     If (E' exists in $G_W$ && $G_S$)
      Add nodes $n_1$ and $n_2$ to $D_f$
      Add edge $e_{n1,n2}$ to $E_f$
      Add weight max ($G_W$ ($w_{n1, n2}$), $G_S$ ($w_{n1, n2}$)) to $W_f$
   Else If (E' exists in $G_W$ || $G_S$)
   Add nodes $n_1$ and $n_2$ to $D_f$
   Add edge $e_{n1,n2}$ to $E_f$
   Add weight $G_W$ ($w_{n1, n2}$) || $G_S$ ($w_{n1, n2}$) to $W_f$
   EndIf
   Remove nodes $n_1$ and $n_2$ in G'
 EndWhile
```

E. Clustering

In this section, the fused graph [21] is portioned into number of sub-graphs to form clusters. Algorithm 4 explains document clustering algorithm.

```
Algorithm 4. Document Clustering
Input: Fused Graph $G_F$= ($D_f$, $E_f$, $W_f$)
Output: Clustered Sub-graph $G_C$ = {$G_1$, G2, G3…$G_k$}
 Reassign each node edges similarity value based
on
$W_f(n_1,n_2)=\frac{CN(n_1 \cap n_2)}{\sqrt{CN(n_1)*CN(n_2)}}$
  Where CN ($n_1$) = neighboring node count of $n_1$
CN ($n_1 \cap$ $n_2$) is intersection of neighboring node
count.
 Add all the nodes to Queue $Q_1$
 Initialize clsId=0;
 While ($Q_1$ is not empty)
  $n_1$=$Q_1$.front()
 Add all neighbors of $n_1$ Queue $Q_2$ if $W_f$ > 0.5
  clsId++;
  While ($Q_2$ is not empty)
    $n_2$=$Q_2$.front()
Find all direct reachable DR nodes from $n_2$ For
each node t in DR
  If node t is not assign to any cluster
  Add node t to clsId cluster
  EndIf
    EndFor
    $Q_2$.pop();
  EndWhile
  $Q_1$.pop();
EndWhile
```

In step1 the all node edges similarity value will be reassigned using $W_f(n_1,n_2)=\frac{CN(n_1 \cap n_2)}{\sqrt{CN(n_1)*CN(n_2)}}$. Select all the neighbors nodes who has the similarity greater than 0.5. Find all nodes that are directly reachable DR [23] from a given node. For each node t in DR, check whether t is already assigned to any cluster or not. If it is not assigned to any cluster, insert t to cluster. This process is repeated until the queue is empty.

III. EXPERIMENTAL RESULTS

The quality of clustering algorithms is frequently based on their performance according to a specific eminence index, in a new evaluation. Experiments either use a incomplete amount of real-world instances or synthetic data. There are two data sets are used for experiments: BBC and 20News Groups.

This paper customs four assessment metrics in turf of the contents in the document retrieval: Purity [24], Precision, Recall and F-measure to experiment the distinction of the proposed clustering algorithm.

To calculate the Purity, clusters are individually allocated to the class where most of the clusters are recursive and then the accurateness of this undertaking is measured by calculating the sum of correctly allocated with the texts and dividing by N [25].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$purity(\Omega, C) = \frac{1}{N}\sum_{k}\max_{j}\left|w_k \cap c_j\right|$$

Where Ω={w1,w2,…,wk} is the set of clusters and C={c1,c2,…cj} is the set of classes.

Precision and Recall can be calculated as,

P = Precision= $\frac{TP}{TP+FP}$ and R= Recall= $\frac{TP}{TP+FN}$

TP = True Positive, TN= True Negative, FP= False Positive, FN= False Negative

The f-measure is calculated as, $F = 2 \times \frac{P.R}{P+R}$

Table I. Dataset Characteristics

Dataset	Docs	Terms	Word Frequency Terms	Semantic Frequency Terms
BBC	1000	10511	9302	8865
20News Groups	1000	11575	9567	10241

Table 1 shows two datasets where the first data set is of BBC, which contains 1000 random documents that belongs to five groups (business, entertainment, politics, sport and tech). The second data set is 20News group, which contains 1000 documents taken randomly which belongs to 10 groups.

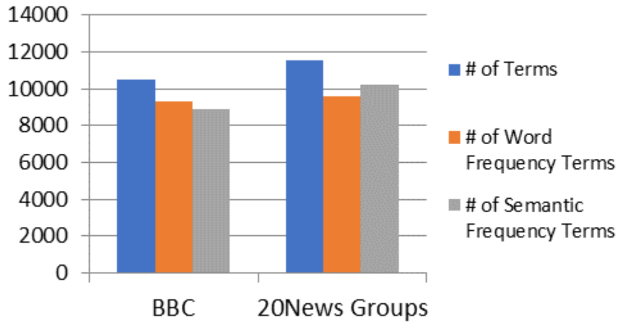

Fig. 3. Dataset Terms, Word and Semantic Frequency

Table II. Evaluation Metrics for Different Graph

Metrics	Word Frequency	Semantic Frequency	**Text Graph**
Purity	0.58	0.66	**0.81**
Precision	0.71	0.79	**0.86**
Recall	0.70	0.76	**0.87**
F-Measure	0.704	0.774	**0.86**

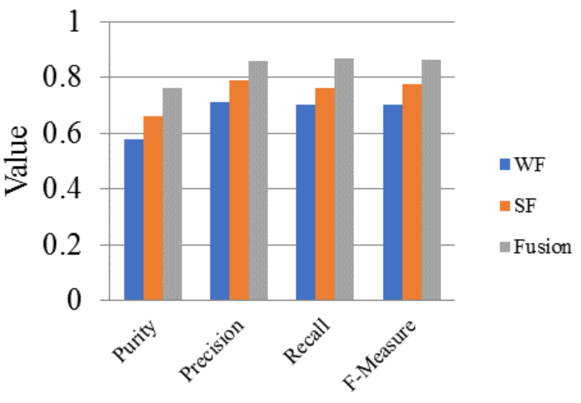

Fig. 4. Comparison of metric with different Graph

Table III. Comparison of Evaluation with Different Algorithms

Algorithm	HC Algorithm	TCABGSSWC	Text Graph
Precision	0.83	0.84	**0.86**
Recall	0.83	0.86	**0.87**
F-Measure	0.831	0.848	**0.864**

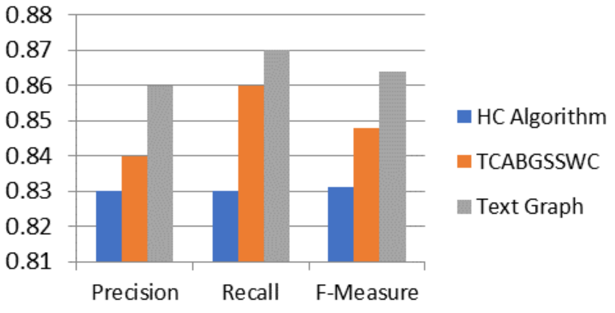

Fig. 5. Comparison of metric with different algorithm

IV. CONCLUSION

On analyzing text graph structure, the proposed paper explains how the text clustering algorithm based on graph structure of fusion model of word frequency and semantic frequency. The TextGraph algorithm uses the graph structure representing the text feature information. The edge value in graph reflects the semantic relationship of two documents. The experimental results depicts the proposed clustering method in this article is very precise and effective using the features extracted in the documents.

REFERENCES

1. Rao, B., & Mishra, S. N.: (2017). An Approach to Text Documents Clustering with {n, n-1... 1}-Word (s) Appearance Using Graph Mining Techniques. *Intl. J. Sci.*

Eng. Adv. Tech., Vol 4(12), pp. 756 -762. (2017).

2. Kaur, R., Kaur, A.: Text Document Clustering and Classification using K-Means Algorithm and Neural Networks. *Ind. J. Sci. Tech.*, Vol 9(40), DOI: 10.17485/ijst/2016/v9i40/9772 2, (2016).
3. Garcia, R., Porrata, A.: Dynamic hierarchical algorithms for document clustering. *Pattern Recognition Letters*, Vol 31 (6), pp 469–477, (2010).
4. Xu, D., Tian, Y.: A comprehensive survey of clustering algorithms. *Annals Data Sci.* Vol 2(2) pp, 165–193, (2015).
5. Jin ,C., Bai, Q.: Text Clustering Algorithm Based on the Graph Structures of Semantic Word Co-occurrence, *Int. Conf. Inf. Sys. Artificial Intelligence*, (2015).
6. Shafiei M.: Document Representation and Dimension Reduction for Text Clustering. *IEEE 23rd Int. Conf. Data Eng. Workshop*, pp. 770-779, , Istanbul, (2007).
7. Win TT, Mon L: Document clustering by fuzzy c-mean algorithm. Advanced computer control (ICACC), *IEEE 2010 2nd Int. Conf.*, Vol 1, pp 239–242 (2010).
8. James CB, Robert E, William F.: FCM: the fuzzy c-means clustering algorithm. *Comput Geosci* 191–203, (1984).
9. Bezdek JC, Pal MR, Keller J, Krishnapuram R.: Fuzzy models and algorithms for pattern recognition and image processing. Kluwer Academic, Massaschusetts (1999).
10. Harish BS, Prasad B, Udayasri B.: Classification of text documents using adaptive fuzzy c-means clustering. Recent advances in intelligent informatics. Springer International Publishing, pp 205–214 (2014)
11. Krishnapuram R, Keller JM.: A possibilistic approach to clustering. *IEEE Trans Fuzzy Syst* 1(2):pp. 98–110. (1993)
12. Tjhi WC, Chen L.: Dual fuzzy-possibilistic co-clustering for categorization of documents. *IEEE Trans Fuzzy Syst* 17(3):pp. 532–543 (2009).
13. Revanasiddappa, M. B., Harish, B. S., Aruna Kumar, S. V.: Clustering Text Documents Using Kernel Possibilistic C-Means, *Proc. Int. Conf. Cognition Recognition*, pp. 127-134 Springer Singapore, (2017).
14. Abualigah, L. M., Khader, A. T., & Al-Betar, M. A.: Multi-objectives-based text clustering technique using K-mean algorithm. *7th Int. Conf. Comp. Sci. Inform. Tech.* (CSIT) (pp. 1–6). IEEE (2016)..
15. Abualigah, L. M., Khader, A. T., Al-Betar, M. A., & Awadallah, M. A.: A krill herd algorithm for efficient text documents clustering. *In IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)* (pp. 67–72). IEEE (2016).
16. V. Tunali, T. Bilgin, and A. Camurcu.: An improved clustering algorithm for text mining: multi-cluster spherical K-means. *Int. Arab J. Inform. Tech.*, vol. 13, no.1, pp. 12–19, 2016.
17. Y. Li, C. Luo, and S. M. Chung.: A parallel text document clustering algorithm based on neighbours. *Cluster Comp.*, vol. 18, no. 2, pp. 933–948, (2015).
18. T. Peng and L. Liu.: A novel incremental conceptual hierarchical text clustering method using CFu-tree. *Appl. Soft Comp.*, vol. 27, pp. 269–278,(2015).
19. X. Pei, T. Wu, and C. Chen.: Automated graph regularized projective nonnegative matrix factorization for document clustering. *IEEE Trans.Cybernetics*, vol. 44, no. 10, pp. 1821–1831, (2014).
20. M. Lu, X.-J. Zhao, L. Zhang, and F.-Z. Li.: Semi-supervised concept factorization for document clustering. *Inform. Sci.*, vol. 331, pp. 86–98, (2016).
21. C.-K. Yau, A. Porter, N. Newman, and A. Suominen.: Clustering scientific documents with topic modeling. *Scientometrics*, vol.100, no. 3, pp. 767–786, (2014).
22. Y. Ma, Y. Wang, and B. Jin.: A three-phase approach to document clustering based on topic significance degree. *Expert Sys. Appl.*, vol. 41, no. 18, pp. 8203–8210, (2014).
23. M.S. Hossain, R.A. Angryk.: Gdclust: A graph-based document clustering technique. *Proc.Seventh IEEE Int. Conf. Data Mining Workshops, IEEE Comp.Soc*,.pp. 417–422 Washington, DC, USA, (2007).
24. Yi, J., Zhang, Y., Zhao, X., and Wan, J.: A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network ,*Hindawi Mathematical Problems in Engineering* Volume (2017).
25. Bharti KK, Singh PK.: Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering . *Appl Soft Comput* 43:pp. 20–34 (2016).

AUTHORS PROFILE

.

M.Uma Maheswari[1], pursuing Ph.D. in Computer Science in the broad domain of Text mining published papers in Graph clustering in Text documents that, Efficient Semantic search through text Repositories and also focuses in Geographical Information Retrieval in Text documents mainly deal with travelogue mining. Further research work is in progress.

Dr. J. G. R. Sathiaseelan[2], Head and Coordinator of Computer Science, Computer Applications and IT Departments in Bishop Heber College, Tiruchirappalli. Dr. J. G. R. Sathiaseelan was awarded Ph. D. degree in Computer Science in the year of 2013. Currently he is guiding 8 Ph.D. Scholars and 5 M.Phil Scholars in full time and part time basis. He has presented several research papers in the International conferences which are published in the proceedings of ACM, IEEE, Elsevier, Inderscience and reputed journals. His research areas include Web Services Security, Big Data Analytics, Data Mining and Image Processing. Dr. Sathiaseelan has authored a book titled, "**Programming In C#.Net**", which was published by PHI, New Delhi, in 2009 and also he has published three more. He is the chairman for both Department Research Committee and the Board of Studies in Bishop Heber College. He is also member of board of studies in Bharathidasan University. He has been frequently invited as chief guest for seminars and conferences both national and international level organized by various colleges in India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication