iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** VI **Month of publication:** June 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

A Survey on Techniques used for Sentence Clustering of Text Documents

Jinto Jacob[#1]

[#]*Department of Computer science, Viswajyothi College of Engineering & Technology, Vazhakulam, Kerala, India*

Abstract— ***Clustering techniques is essential in the data mining process to reveal natural structures and identify interesting patterns in the underlying data. Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups. Fuzzy clustering algorithms allow patterns to belong to all clusters with differing degrees of membership. This is important in domains such as sentence clustering, since a sentence is likely to be related to more than one theme or topic present within a document or set of documents. However, because most sentence similarity measures do not represent sentences in a common metric space, conventional fuzzy clustering approaches based on prototypes or mixtures of Gaussian are generally not applicable to sentence clustering. Some of the clustering algorithms are taken here for literature survey. The survey compared these methods and identified the problems in the existing systems. A very rich literature on cluster analysis has developed over the past three decades. Many conventional clustering algorithms have been adapted or directly applied to text data, and also new algorithms have recently been proposed specifically aiming at text data. This survey discuss about the different clustering algorithm and similarity measures available. Different problems of current system are also identified. Finally propose a new model for fuzzy clustering of sentence data.***

Keywords— ***Fuzzy clustering, Clustering algorithm.***

I. INTRODUCTION

This world is full of data. Every day, people encounter a large amount of information and store or represent it as data, for further analysis and management. One of the vital means in dealing with these data is to classify or group them into a set of categories or clusters. Clustering algorithms partition data into a certain number of clusters (groups, subsets, or categories). Most researchers describe a cluster by considering the internal homogeneity and the external separation i.e., patterns in the same cluster should be similar to each other, while patterns in different clusters should not. Both the similarity and the dissimilarity should be examinable in a clear and meaningful way.

Sentence clustering plays an important role in many text processing activities. For example, various authors have argued that incorporating sentence clustering into extractive multi-document summarization helps avoid problems of content overlap, leading to better coverage. However, sentence clustering can also be used within more general text mining tasks. For example, consider web mining, where the specific objective might be to discover some novel information from a set of documents initially retrieved in response to some query. Clustering the sentences of those documents would intuitively expect at least one of the clusters to be closely related to the concepts described by the query terms; however, other clusters may contain information pertaining to the query in some way unknown. If the information in such data are found it would successfully mined new information.

Irrespective of the specific task most documents will contain interrelated topics or themes, and many sentences will be related to some degree to a number of these. The successful capture of such fuzzy relationships will lead to an increase in the breadth and scope of problems to which sentence clustering can be applied. However, clustering text at the sentence level poses specific challenges not present when clustering larger segments of text, such as documents.

Clustering text at the document level is well established in the Information Retrieval (IR) , where documents are typically represented as data points in a high dimensional vector space in which each dimension corresponds to a unique keyword, leading to a rectangular representation in which rows represent documents and columns represent attributes of those documents (e.g., tf-idf values of the keywords). This type of

data is amenable to clustering by a large range of algorithms. Since data points lie in a metric space, prototype-based algorithms can be applied, which represent clusters in terms of parameters such as means and covariance, and therefore assume a common metric input space.

The works done in the paper deals with different approaches and similarity measure used for clustering. The main issues in the clustering are the noise in data, high dimensionality, etc,. The survey starts with a clustering algorithm. The second and third works will deal with some ranking algorithm. These ranking algorithms can be used in the proposed method for ranking the keywords. The fourth survey work deals with text summarization. In this work provide a ranking of sentence and extraction of the top sentences. It also provides different measure for ranking sentences. The fifth work is a fuzzy clustering algorithm which is based on the c-means algorithm. The sixth survey work provides a comparison of different similarity measures. The work in seven and ten are summarization of sentence which uses the ranking of the sentence and can be also used for ranking keywords. The eighth work in the survey provides a clustering algorithm. The ninth survey work is a keyword extraction method for clustering. This work provides a clear framework for the keyword extraction and compares different ranking methods provided.

II. Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering

A novel method for simultaneous key phrase extraction and generic text summarization [2] is proposed for modeling text documents as weighted undirected and weighted bipartite graphs. The goal of text summarization is to take a textual document, extract content from it and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs. This method adopt the unsupervised approach. It explicitly model both keyphrases and the sentences that contain them using weighted undirected and weighted bipartite graphs and generate sentence extracts on the fly without extensive training.

For each document, the method generate two sets of objects: one the set of terms $T = \{t_1,\ldots,t_n\}$ and the other the set of sentences $S = \{s_1,\ldots,s_m\}$ in the document and build a weighted bipartite graph from T and S in the following way: if the term ti appears in sentence s_j , then create an edge between t_i and s_j . Non negative weight can also be given to the edges of the weighted bipartite graph with w_{ij} indicating the weight on the edge (t_i, s_j). This weighted bipartite graph is represented by G(T, S, W) where $W = [w_{ij}]$ is the m-by-n weight matrix containing all the pair-wise edge weights. For each term ti and each sentence sj compute their saliency scores $u(t_i)$ and $v(s_j)$, respectively. The saliency score of a term is determined by the saliency scores of the sentences it appears in, and the saliency score of a sentence is determined by the saliency scores of the terms it contains.

Now collect the saliency scores for terms and sentences into two vectors u and v, respectively. The terms and sentences are ranked in decreasing order of their saliency scores, and select the top t terms (with the highest saliency scores) to add to the top term list and the top s sentences (with the highest saliency scores) to add to the summary.

Disadvantage

1. More research needed to find optimal link strength for this method.

III. A New Fuzzy Relational Clustering Algorithm Based On The Fuzzy C-Means Algorithm

This paper proposes a new fuzzy relational algorithm [6], based on the popular fuzzy C-means (FCM) algorithm, which does not require any particular restriction on the relation matrix. In fuzzy relational clustering, the problem of classifying data is solved by expressing a relation that quantifies the similarity, or dissimilarity, degree between pairs of objects. Based on such relation, objects very similar to each other, i.e., objects of the same type will belong with high membership values to the same cluster. This algorithm takes the FCM algorithm as starting point. FCM is an iterative algorithm which partitions a data set minimizing the Euclidean distance between each point (strongly) belonging to a cluster and the prototype of the cluster. This is obtained by updating at each iteration both the membership of each point to a cluster and the cluster prototypes.

In FCM, a prototype is a point which is representative of the cluster and has a strategic position with respect to the neighboring. In ARCA each object is represented by the vector of its relation strengths with the other objects in the data set, and a prototype is an object whose relationship with all the objects in the data set is representative of the mutual relationships of a group of similar objects. Like FCM, ARCA

partitions the data set minimizing the Euclidean distance between each object (strongly) belonging to a cluster and the prototype of the cluster.

In FCM, a prototype is a point which is representative of the cluster and has a strategic position with respect to the neighboring objects. In ARCA each object is represented by the vector of its relation strengths with the other objects in the data set, and a prototype is an object (possibly not included in the original data set) whose relationship with all the objects in the data set is representative of the mutual relationships of a group of similar objects. ARCA partitions the data set minimizing the Euclidean distance between each object (strongly) belonging to a cluster and the prototype of the cluster.

IV. Constrained Text Coclustering with Supervised and Unsupervised Constraints

This work proposes a novel constrained coclustering method [8] to achieve two goals. First, combine information theoretic coclustering and constrained clustering to improve clustering performance. Second, it adopts both supervised and unsupervised constraints to demonstrate the effectiveness of the algorithm. The unsupervised constraints are automatically derived from existing knowledge sources, thus saving the effort and cost of using manually labeled constraints. To achieve the first goal, it uses a two-sided hidden Markov random field (HMRF) model to represent both document and word constraints. It then uses an alternating expectation maximization (EM) algorithm to optimize the model. The method also propose two novel methods to automatically construct and incorporate document and word constraints to support unsupervised constrained clustering: 1) automatically construct document constraints based on overlapping named entities (NE) extracted by an NE extractor; 2) automatically construct word constraints based on their semantic distance inferred from WordNet. The document set and word set is denoted as D and V Then the joint probability of p(dm; vi) can be computed based on the co-occurrence count of dm and vi.

There are two steps in the EM algorithm: the E-step and the M-step. The E-Step update the cluster labels based on the fixed model function from the last iteration. More exactly, use the iterated conditional mode (ICM) algorithm to find the cluster labels.

ICM greedily solves the objective function by updating one latent variable at a time, and keeping all the other latent variables fixed. The M-Step update the model function by fixing Ld and Lv. Since the latent labels are fixed, the update of q is not affected by the must-links and cannot-links.

Advantage

1. It performed better than the existing coclustering algorithms because it allows the system to incorporate additional constraints to guide the clustering towards the ground-truth.

2. It performed better than the existing 1D constrained clustering methods since it can take advantage of the cooccurrences of documents and words;

3. It performed better than the existing constrained coclustering approaches on text data since it optimizes a KL-divergence based objective function versus a Euclidean distance-based function that is commonly used by other systems.

V. Analysis of Statistical Keyword Extraction Methods for Incremental Clustering

The different keyword extraction methods have different assumptions about the properties of the keywords, which end up with different sets of keywords extract by the different methods. Thus, an analysis about which method is more appropriate for the incremental clustering task is necessary. Besides, a study about the number of keywords to improve or maintain the quality of the incremental clustering is also necessary, since the use of a little number of keyworks might not maintain the quality of the incremental clustering and a large number of keywords might not have impact in the speed of the process. Then, this method aims to analyze different methods for keyword extraction and analyze the impact of the different number of keywords extracted from documents in the quality of the incremental clustering.

The statistical keyword extraction methods analyzed in this work are based on a sentence-term matrix. In this matrix, each row corresponds to a sentence of a document and each term corresponds to a column. A term can be a single word or a set/sequence of words. The set of terms are denoted as T = {t1, t2,…, tN} and the set of sentences as S = {s1, s2,…, sM} in which sj ϵ T . If a term ti occurs in a sentence sj , the value 1 is assigned to the corresponding cell of the matrix (oti , sj) and 0 otherwise.

This work defines a framework for statistical keyword extraction. The framework is defined in 5 steps: i) preprocess the textual document, ii) generate the sentence-term matrix, iii) generate scores for each term extracted from the document, iv) sort the term scores, and v) extract the first k terms of the sorted term scores as keywords.

The keyword extraction methods are given below.

A. Most Frequent

A simple measure to automatically extract keywords is to consider the most frequent terms as keywords. The score of the term ti is obtained by counting the number of occurrences of the term in the sentence-term matrix.

B. Term Frequency - Inverse Sentence Frequency

The basic idea of TF-ISF (Term Frequency - Inverse Sentence Frequency) measure is to determine the score of a term according to its frequency and its distribution through the sentences of the document. The score of a term decreases if a term occurs in a large number of sentences in the document, since this can be a common term and do not characterize the content of the document.

C. Co-occurrence Statistical Information

CSI (Co-occurrence Statistical Information) measure obtain scores for words using $\chi 2$ measure . $\chi 2$ measures how much the observed frequencies are different from the expected frequencies.

D. Eccentricity-Based

Eccentricity is a centrality measure, i.e., a measure which determines the importance of a node in a graph . According to eccentricity measure, a node is central if its distance to the most distance node is small. The distance between a term ti and term tj ie, d(ti; tj) is given by the sum of the edge weights on the shortest path from ti to tj in G.

E. TextRank

TextRank [4] algorithm is based on PageRank algorithm, which defines the importance of a vertices in the graph considering the importance of its connected objects.

From the above analysis the most suitable keyword extraction method is identified as the most frequent in case of the non graph based methods and TexRank in the graph based method. We can use them according to the need of user.

VI. Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm

This is a method provided by Skabar et al. [1] for clustering of sentence in fuzzy manner, ie, one sentence can belong to more than one cluster at the same time. The method is given as below.

The contribution of this work [1] is a novel fuzzy relational clustering algorithm. Inspired by the mixture model approach, which model the data as a combination of components. However, unlike conventional mixture models, which operate in a Euclidean space and use a likelihood function parameterized by the means and covariances of Gaussian components, use of any explicit density model (e.g., Gaussian) for representing clusters is abandoned. Instead of that a graph representation in which nodes represent objects, and weighted edges represent the similarity between objects is used. Cluster membership values for each node represent the degree to which the object represented by that node belongs to each of the respective clusters, and mixing coefficients represent the probability of an object having been generated from that component. By applying the PageRank algorithm to each cluster, and interpreting the Page-Rank score of an object within some cluster as a likelihood, the Expectation-Maximization (EM) framework can be used to determine the model parameters.

The result is a fuzzy relational clustering algorithm which is generic in nature, and can be applied to any domain in which the relationship between objects is expressed in terms of pair-wise similarities.

The PageRank is used for similarity. The underlying assumption for calculating the importance of a sentence is that sentences which are similar to a large number of other important sentences are central. A commonly used measure to assess the importance of the words in a sentence is the inverse document frequency, or idf.

The similarity between two sentences is defined by cosine similarity. All the numbers are normalized so that the highest ranked sentence gets the score 1. This is then taken as similarity matrix and given to maximization algorithm.

Unlike a web graph, in which edges are unweighted, edges on a document graph are weighted with a value representing the similarity between sentences.

The proposed algorithm uses the PageRank score of an object within a cluster as a measure of its centrality to that cluster. These PageRank values are then treated as likelihoods. Since there is no parameterized likelihood function as such, the only parameters that need to be determined are the cluster membership values and mixing coefficients. The algorithm uses Expectation Maximization to optimize these parameters.

This algorithm works in 3 steps as Initialization ,Expectation and Maximization .

At initialization step cluster membership values are initialized randomly, and normalized such that cluster membership for an object sums to unity over all clusters. Mixing coefficients are initialized such that priors for all clusters are equal. An alternative to random initialization is to initialize cluster membership values with values found by first applying a computationally inexpensive hard clustering algorithm such as Spectral Clustering or k-Medoids. This will result in each object having an initial membership value of either 0 or 1 to each cluster. In practice it have a significant effect on the rate of convergence, with convergence typically achieved in 30 to 50 EM cycles approximately one tenth the number of iterations required when using random initialization

The Expectation step calculates the PageRank value for each object in each cluster. Once PageRank scores have been determined, these are treated as likelihoods and used to calculate cluster membership values. The maximization step involves only the single step of updating the mixing coefficients based on membership values calculated in the Expectation Step.

The output of the algorithm is in the form of a matrix of size s*m where s is number of the sentence and m is number of clusters. Each entry will be the membership value of the sentence in the cluster. There is two kind of clustering possible using this matrix a fuzzy clustering, and hard clustering. In fuzzy clustering the sentences are assigned to all clusters in which its membership value is above a threshold allowing some sentence to be in multiple clusters. In hard clustering the sentence is assigned to cluster for which membership value is highest.

This algorithm is able to find out overlapping clusters in semantic sentences. Potential application of the algorithm is document summarization and text mining. Like any clustering algorithm, the performance of FRECCA will ultimately depend on the quality of the input data, and in the case of sentence clustering this performance may be improved through development of better sentence similarity measures, which may in turn be based on improved word sense disambiguation, etc. Any such improvements are orthogonal to the clustering model, and can be easily integrated into it.

FRECCA is not sensitive to the initialization of cluster membership values. The algorithm appears to be able to converge to an appropriate number of clusters, even if the number of initial clusters was set very high. The algorithm can also be applied to asymmetric matrix. It can also be applied to attribute data. This might be done by first calculating pairwise distances between pairs of attribute vectors using some suitable distance measure (e.g., euclidean, Mahalanobis, etc.), and then converting these distances to similarities by passing them through a suitable monotone decreasing function.

VII. CONCLUSIONS

As an important tool for data exploration, cluster analysis examines unlabeled data, by either constructing a hierarchical structure, or forming a set of groups according to a pre-specified number. This work focuses on the clustering algorithms and reviews a wide variety of approaches appearing in the literature. Usually, algorithms are designed with certain assumptions and favor some type of biases. In this sense, it is not accurate to say “best” in the context of clustering algorithms, although some comparisons are possible. At the preprocessing and post-processing phase, feature selection and cluster validation can improve the efficiency of clustering algorithms.

REFERENCES

[1] Andrew Skabar, Khaled Abdalgader, "Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm", in IEEE Transactions on Knowledge And Data Engineering, vol. 25, no. 1, January 2013..

[2] J. Hongyuan Zha, "Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering", in Proc. 25th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval , pp. 113-120, 2002. 2002.

[3] S. G. Erkan and D.R. Radev, "LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization," in J. Artificial Intelligence Research, vol. 22, pp. 457-479, 2004.

[4] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in Proc. Conf. Empirical Methods in Natural Language (EMNLP), pp. 404-411, 2004.

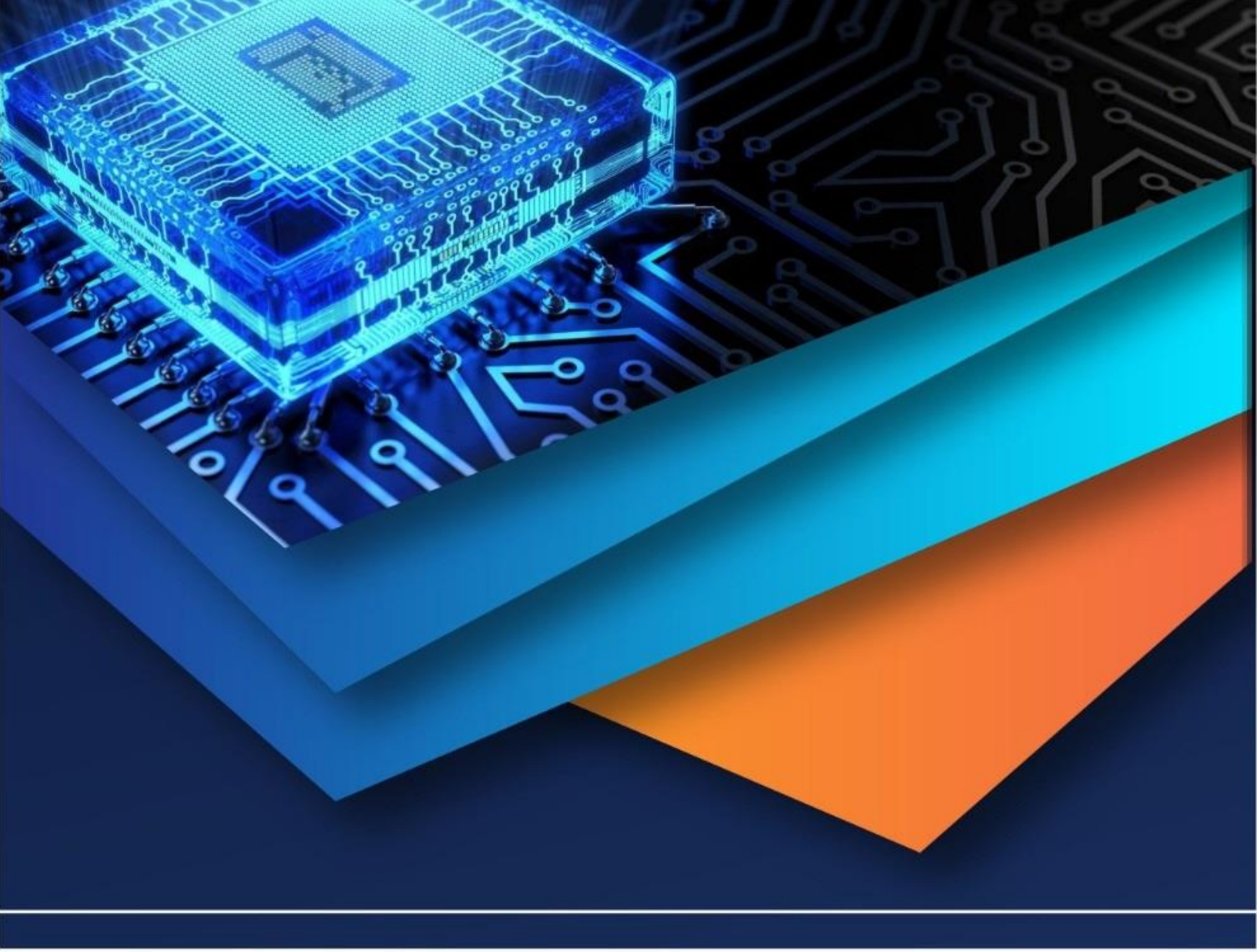
[5] D.R. Radev, H. Jing, M. Stys, and D. Tam, "Centroid-Based Summarization of Multiple Documents," in Information Processing and Management: An Int'l J., vol. 40, pp. 919-938, 2004.

[6] P. Corsini, F. Lazzerini, and F. Marcelloni, "A New Fuzzy Relational Clustering Algorithm Based on the Fuzzy C-Means Algorithm," in Soft Computing, vol. 9, pp. 439-447, 2005.

[7] D. Wang, T. Li, S. Zhu, and C. Ding, "Multi-Document Summarization via Sentence- Level Semantic Analysis and Symmetric Matrix Factorization," in Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval ,pp.307-314,2008.

[8] Yangqiu Song, Shimei Pan, Shixia Liu, Michelle X. Zhou, Weihong Qian, "Constrained Text Coclustering with Supervised and Unsupervised Constraints", in IEEE Transactions on knowledge and data engineering, vol. 25, no. 6, june 2013.

[9] Rafael Geraldeli Rossi, Ricardo Marcondes Marcacini, Solange Oliveira Rezende, "Analysis of Statistical Keyword Extraction Methods for Incremental Clustering", 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY