HIERARCHAL CLUSTERING AND SIMILARITY MEASURES ALONG WITH MULTI REPRESENTATION

A.LAKSHMI DEEPTHI[1], J.V.D PRASAD[2]

[1] Student, [2] Sr. Asst. Professors, Department of CSE, VR Siddhartha Engineering College
556lucky@gmail.com, prasadj@vrsiddhartha.ac.in

Abstract

All clustering methods have to assume some cluster relationship on the list of data objects that they really are applied on. Graph-Based Document Clustering works with frequent senses rather than frequent keywords used in traditional text mining techniques.Similarity between a pair of objects can be defined either explicitly or implicitly. With this paper, we analyzed existing multi-viewpoint based similarity measure and two related clustering methods. The main difference between a traditional dissimilarity/similarity measure and ours could be that the former uses merely a single viewpoint, which is the origin, even though the latter utilizes many viewpoints, which you ll find are objects assumed to not have the very same cluster using the two objects being measured. Using multiple viewpoints, more informative assessment of similarity could well be achieved. Theoretical analysis and empirical study are conducted to back up this claim. Two criterion functions for document clustering are proposed dependent on this wonderful measure. We compare them several well-known clustering algorithms which use other popular similarity measures on various document collections confirming the good sides of our proposal.

Keywords –*Multiview Cluster, Document id, ClusterDistance*

---*---**

1. INTRODUCTION

In GDClust, we construct document-graphs from text documents and apply an Apriori paradigm for locating frequent sub graphs from them. We utilizea hierarchic representation of English terms, WordNet [1], to construct document-graphs Since each document might be represented as graph of related terms, they could be searched for frequent sub graphs using graph mining algorithms. We aim to cluster documents based on the similarity of one's sub graphs in the document-graphs. GDClust enables clustering of documents providing humanlike sense-based searching capabilities, rather than focusing only on the co occurrence of frequent terms. It is sensible the processes by which human beings process the text data.

[2] Proposed well-known sub graph discovery systems like FSG (Frequent Sub graph Discovery), Span (graph-based Substructure pattern mining) , DSPM(Diagonally

Sub graph Pattern Mining) [1], and SUBDUE. These works allow us to believe the fact that the thought of construction of document-graphs and discovering frequent sub graphs to obtain sense-based clustering our effort is feasible. Each one of these systems encounters multiple aspects of efficient frequent sub graph mining. Most of them could have been tested on real and artificial datasets of chemical compounds. Not anyone has actually been applied however, to mine the text data. Within this paper, we discuss GDClust that performs frequent sub graph discovery from text repository in the goal document clustering

Hierarchical clustering showing relations amongst the individual objects and merging clusters files in accordance to similarity along with multi representation. There are a couple of types of hierarchical clustering methods. Agglomerative get started by some part and recursively add two or more appropriate clusters. It Stop when k wide range of clusters is achieved. Hierarchal agglomerative clustering, beginning with all instances inside their own cluster Until there is always one unit cluster Assumes a similarity functions for determining the similarity of two instances Here input is dataset first find the keywords typically from a document and retrieves corresponding words from WorldNet, but we can locate the similarity between two objects in accordance to different viewpoints. Using hierarchical document clustering to get, better cluster quality, high dimensionality, large-scale hard drive data recovery, ease in browsing, and meaning full cluster labels. Reduces the high false positive rate

Document clustering or Text categorization is related to reasoning behind data clustering. Document clustering is basically a more specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering. By way of example, as quantity of online information is increasing rapidly, users along with Information retrieval system had the need to classify the desired document against a specific query. Generally two kinds of clustering approaches are used where one is bottom up and the second kind is top down. With this paper you can find centered on overall performance K-means clustering algorithm, a top down clustering algorithm which assigns each document to the cluster whose center (also termed as centroid) is nearest. Here the documents are represented in vector space model as document vector and of course the center is the average of most the documents in the cluster.

2. LITERATURE SURVEY

[1]A phrase-based document similarity is presented with this paper. By mapping each node of a suffix tree (excludes the main node) into your unique dimension relevant to an M-dimensional term space (M would be the total number of nodes except the fundamental node), each document is represented by way of a feature vector of M nodes. Consequently, we find a basic technique to compute the document similarity: First, the excess body fat (tf-idf) of each and every node is recorded in building the suffix tree, probably the cosine similarity measure is made use of to compute the pair wise similarities of documents. By putting on the brand new document similarity towards the group-average HAC algorithm (GHAC), we made a new document clustering approach. For Entropy, that is used to count how various kinds of documents are distributed within each cluster, the typical score is 0.079.

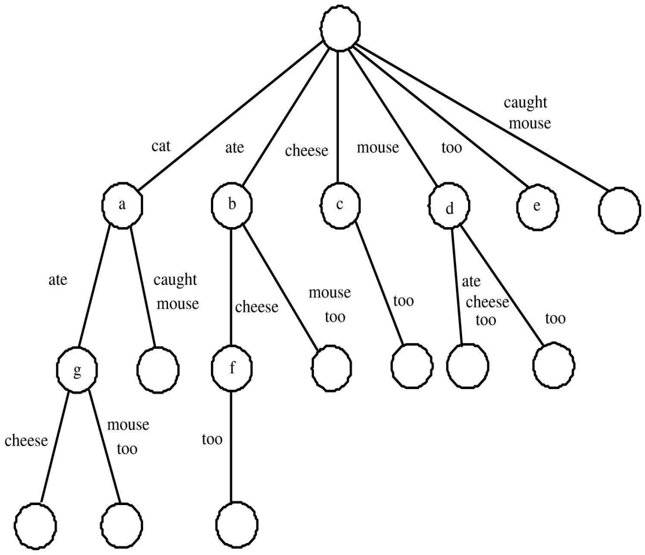

Fig: The suffix tree of four documents after inserting a document “cat caught mouse.”

ADVANTAGES:

Very simple to extract exact documents information. High document clustering rate. Improved cosine Single word similarity measure. Since the period of a word to the wise is variable, it is quite difficult to include a suffix tree dependent on words directly. To deal with the challenge, we create wordlist to accumulate all keywords in alphabetical order.

DISADVANTAGES:

Each document becomes an array of word_ids for your suffix tree construction.

More text parsing time less performance using F-measure Problem in identifying and extracting the phrases in documents

[2]Graph query refinement method proposed by Tomita et al. Our bodies depend upon user interaction when it comes to the hierarchic organization of a text query. In contrast, we depend on a predefined ontology, for automated retrieval of frequent sub graphs from text documents. GDClust gives a fully automated system which uses Apriority-based sub graph discovery technique to harness the possible of sense-based document clustering.

Document-graph construction algorithm selects informative keywords given by a document and retrieves corresponding words from WorldNet. Then, it traverses about the topmost level of abstraction to find all related abstract terms and also their relations. The graph of this very links between keywords’ sunsets for every document and their abstracts compose the individual document-graph. The 1000 documents were chosen from 10 different groups of 20-newsGroup Dataset.

ADVANTAGES:

Effective Document relationship using association algorithm and Graph based approach. Graph level wise filtering using threshold. Improving the efficiency of Apriority algorithm, we used hash-based technique. Dynamic support assignment

DISADVANTAGE:

Inexact matching will allow us to decide on only larger sub graphs created by the Apriori approach that could farther decrease computational costs involved in the phase of frequent sub graph candidate analysis. Document cluster performance suffers by varying support threshold.

[3] Propose a new hierarchical document clustering method that puts together the merits of agglomerative and partition clustering algorithms, but without dropping any words like for example frequent itemset clustering. They first pay a partitioning clustering method of find the initial clusters then apply an agglomerative method of design a hierarchy. This hybrid approach takes some great benefits of partitioning approach for efficiently handling large number of documents, and agglomerative approach of building hierarchy. The usual partitioning approach, such as k-means, creates a flat clustering solution. Though cluster quality is useful, it does not facilitate browsing. Contrastingly, the output of their total

clustering algorithm is naturally a hierarchy of clusters that facilitates effective browsing.

ADVANTAGES:

Clustering algorithm consists of two phases. First, we employ the partitioning clustering way for you to group the document objects into a great deal of clusters. Then, they use agglomerative hierarchical clustering strategy to merge clusters based on their inter-connectivity and closeness. This hybrid approach takes the greatest advantage here of one's efficient and scalable nature from partitioning method whenever the wide range of (document) objects is big, and the benefit for the easy browsing hierarchical structure from hierarchical clustering method. Right here is the secret for achieve efficiency and scalability in your method. Our hybrid method utilizes all items (words) of this very document set and avoids the sensitivity into the minimum support as in frequent itemset clustering method.

Kalman filtering is made use of to calculate internal closeness and internal inter-connectivity of the clusters which remove outliers. The Kalman filter is undoubtedly an efficient recursive filter that estimates the condition of a powerful system typically from a a number of incomplete and noisy measurements.

DISADVANTAGES:

Can’t distinguish the phrases inside the documents. Document dataset limitation under 10kb.High False positive rate.

[4] K-means is based on the objective to cluster n documents based on terms into k partitions so the intra-document similarity is high rather than inter-document similarity. However, the clustering performance of this very K-means algorithm is dependent upon the primary exploration of the centroid point for the cluster. These centroids should be placed in a cunning way because of different location causes different result. So, more suitable choice is to place them whenever you can distant from each other. Yet in case of simple Kmeans algorithm we have now revealed that for the initial consideration of the centroid are performed randomly. So occasionally may produce very poor performance since it fails to classify the fax in disjoint sets.

In this proposed a powerful technique to measure the 1st guess for the centroid points for K clusters. Here the fax are represented in the vector space model and several dissimilarity measurement techniques can easily be applied in the document set to find out the most dissimilar K documents. We've utilized the Jaccard distance measure for locating the K most dissimilar documents. Then these K points should be used as K centroid which guarantees to classify the document in K disjoint sets.

ADVANTAGES:

This product retrieved an arrangement tokens by removing non relevant features that occur uniformly across all documents among the corpus. We have seen that many of words secure the canonical form of morphologically rich syntactic categories, like nouns or verb. For it we've used Suffix Porter’s Stemming algorithm. The error rate of stemming are measured around 5%.Frequency based feature selection provides significance performance in text categorization. This feature selection procedure is based upon the thought that relevant feature will probably be selected which you will find are free form local minima problem.

DISADVANTAGES:

- More clustering error.
- Doesn’t handle supervised dataset.
- Static k value in improved kmeans.

[5] Multi-viewpoints based Similarity the cosine similarity calculating the cosine angle between two document vectors as measuring them at the origin i.e vector 0. Hence, this is actually a single viewpoint-based measure. The motivation of MVS stands it is more than possible acquire a more accurate assessment of how close or distant the document points (di and dj) is, should we could measure them by waiting on in excess of only 1 viewpoint as references. Just for example, given by a third point dh, the direction and distances to di and dj are indicated by two new vectors (di – dh) and (di – dh) respectively. Therefore, working on different vectors with a range of different viewpoints

ADVANTAGES:

The Euclidean distance between objects to its cluster center should really be minimized, as the cosine similarity between them should be maximized. While most of viewpoints are of help, there could be a number of them giving misleading information. Therefore, it suggests a considerable enough number of viewpoints is frequently needed to balance and overcome the effect of misleading viewpoints. In such cases, in case the bigger number of them will certainly be useful, a more informative similarity could well be offered than the single origin point based similarity measure. means the number of the smallest class size to the largest class size within the particular dataset. All datasets are extremely unbalanced except for classic. They had been all preprocessed by standard procedures, including stop-word removal, stemming, removal of too rare along with too frequent words, weighting and normalization.

DISADVANTAGES:

- Supports just for spherical brand of clusters. Doesn’t handle fully supervised document datasets.
- Normalized Mutual Information (NMI) measures the

data the true class partition and of course the cluster assignment share.

- Less NMI rate after document clustering.

CONCLUSIONS AND FUTURE SCOPE

In this particular paper, we studied the traditional Multiviewpoint-based Similarity measuring methods, named MVS. MVS is potentially more desirable for text documents when compared to the popular cosine similarity. Clustering methods that use many kinds of similarity measure, on a lot of of document data sets and under different evaluation metrics, the proposed algorithms demonstrate that might also provide significantly improved clustering performance. Future methods could make use of the same principle, but define alternative forms for your relative similarity. Here we concentrates on partitional clustering of documents. In the future, it could even be a possibility applies the proposed criterion functions for hierarchical clustering algorithms. Finally, we've shown the appliance of MVS and its clustering algorithms for text data. It may be interesting to explore the way how they can work on other types of sparse and high-dimensional data. In future we will extend the work to search documents using mvc on different types of files.

REFERENCES

[1] Efficient Phrase-Based Document Similarity for Clustering Hung Chim and Xiaotie Deng, Senior Member, IEEE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 20, NO. 9, SEPTEMBER 2008.

[2] GDClust: A Graph-Based Document Clustering Technique M. Shahriar Hossain, Rafal A. Angryk, Seventh IEEE International Conference on Data Mining – Workshops.

[3] An Efficient Hybrid Hierarchical Document Clustering Method Yehang Zhu, Fifth International Conference on Fuzzy Systems and Knowledge Discovery.

[4] An efficient K-Means Algorithm integrated with Jaccard Distance Measure for Document Clustering Mushfeq-Us-Saleheen Shameem, 2009 IEEE.

[5] Gerald Kowalski, “Information Retrieval Systems – Theory and Implementation”, Kluwer Academic Publishers, 1997.

[6] Cutting, D. R.; Karger, D.; Pedersen, J.; and Tukey, J. W. 1992. “Scatter/Gather: A cluster-based approach to browsing large document collections “. In Proceedings of SIGIR-92. pp. 318–329. Copenhagen, Denmark.