www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 11 Nov 2015, Page No. 14921-14923

Text clustering for computer investigation with search optimization

Jameer Kotwal, Prajkta Varhade, Komal Koli, Roshni Gawande, Varsha Andhale

[1] N.M.I.E.T. Department of computer, Pune university
Talegaon Dabhade
jameerktwl@gmail.com

[2] N.M.I.E.T. Department of computer, Pune university
Talegaon Dabhade
prajktavarhade@gmail.com

[3] N.M.I.E.T. Department of computer, Pune university
Talegaon Dabhade
Komalkoli1993@gmail.com

[4] N.M.I.E.T. Department of computer, Pune university
Talegaon Dabhade
Roshnigawande84@gmail.com

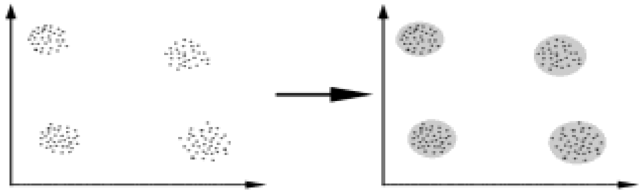
[5] N.M.I.E.T. Department of computer, Pune university
Talegaon Dabhade
Varshaandhale8@gmail.com

Abstract: Recently, in the world of digital technologies especially in computer world, there is a tremendous increase in crimes. So investigation of such cases deserves a much more importance. Computer Investigation is the process of uncovering and interpreting process of electronic data for use in computer forensic analysis. Text clustering is an automatic grouping of text documents into clusters so that documents within a cluster have high similarity in comparison to one another, but are dissimilar to documents in other clusters. We are carrying out extensive experimentation with two well-known clustering algorithms K-means and Hierarchical single/Complete/Average Link. In this the number of unstructured text files is given as input and the output is generated in structured format. Previous experiments have been performed with different combinations of algorithms and parameters. Related studies in the literature are significantly more limited than our study. Our studies show that the K-means and Hierarchical algorithms provide the best results for our application domain.

Keywords: Text Clustering, clustering algorithm, Forensic analysis

1. Introduction

What is Clustering?
Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups.



It is measuring similarity between documents and grouping similar documents together. It provides efficient representation and visualization of the documents. We present an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigations. In our particular application domain, it usually involves examining hundreds of thousands of files per computer. In this methods for automated data analysis, like those widely used for machine learning and data mining are of predominant importance. In this context, the use of clustering algorithms, which are capable of finding required data from text documents found in seized computers. In previous datasets there were unlabeled objects. The classes or categories of documents that can be found are *a priori* unknown. Our system provide labeled search.

2. K-means Algorithm

An algorithm for partitioning (or clustering) N data points into K disjoint subsets S_j containing data points so as to minimize the sum-of-squares criterion.

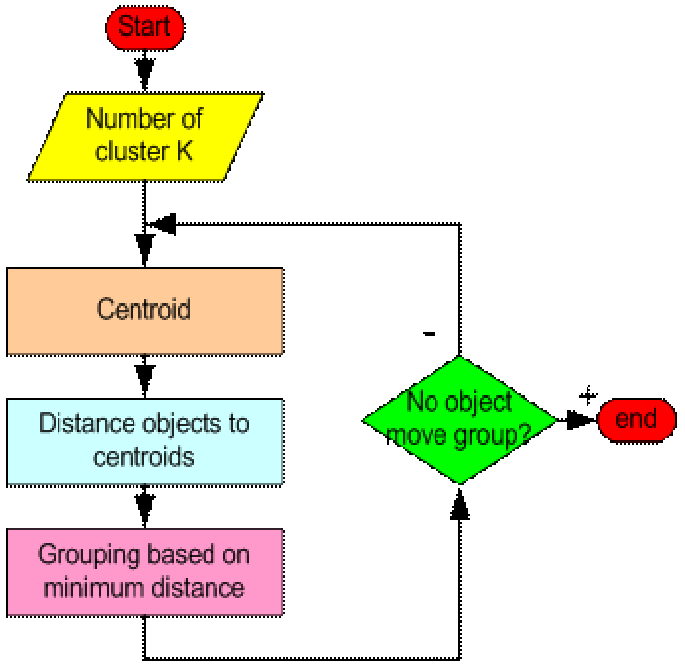
$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2,$$

where x_n is a vector representing the the n^{th} data point and u_j is the geometric centroid of the data points in S_j.
The k-means algorithm is an algorithm to cluster *n* objects based on attributes into *k* partitions, where *k* < *n*.

- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group. K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the

corresponding cluster centroid.

How K-means algorithm works:


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

- *K-means algorithm is* useful for undirected knowledge discovery and is relatively simple. K-means has found wide spread usage in lot of fields, ranging from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligence, image processing, machine vision, and many others.

3. Hierarchical Algorithm

These find successive clusters using previously established clusters.

1.Agglomerative("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

- Assumes a *similarity function* for determining the similarity of two instances.
- Starts with all instances in a separate cluster and then repeatedly joins the two clusters that are most similar until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

2.Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

*Single Link: Similarity of two most similar members.

- Use maximum similarity of pairs:
- $$sim(c_i,c_j) = \max_{x \in c_i, y \in c_j} sim(x,y)$$
- Can result in “straggly” (long and thin) clusters due to *chaining effect*.

*Complete Link: Similarity of two least similar members.

- Use minimum similarity of pairs:

$$sim(c_i,c_j) = \min_{x \in c_i, y \in c_j} sim(x,y)$$

- Makes more “tight,” spherical clusters that are typically preferable.

*Group Average: Average similarity between members.

- Use average similarity across all pairs within the merged cluster to measure the similarity of two clusters.

$$sim(c_i,c_j) = \frac{1}{|c_i \cup c_j|(|c_i \cup c_j| - 1)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j): \vec{y} \neq \vec{x}} sim(\vec{x},\vec{y})$$

- Compromise between single and complete link.
- Averaged across all ordered pairs in the merged cluster instead of unordered pairs *between* the two clusters (to encourage tighter final clusters).

4. Architecture

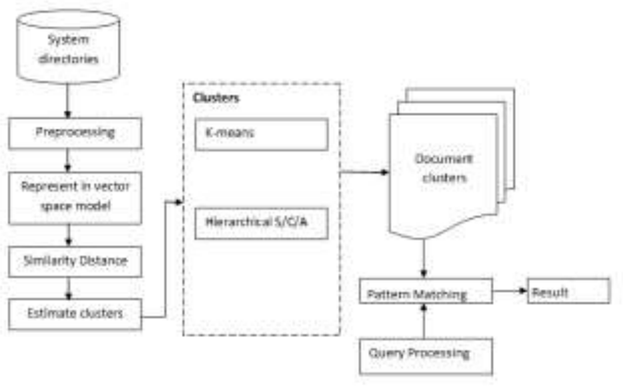


Fig. Architecture Diagram

A) Preprocessing:

The removal of stop words is the most common term filtering technique used. There are standard stop word lists available but in most of the applications these are modified depending on the quality of the dataset. Stemming is the process of reducing words to their stem or root form. For example ‘cook’, ’cooking’, ‘cooked’ are all forms of the same word used in different constraint but for measuring similarity these should be considered same.

B) Vector Space Model:

Frequency for each word will get count and that will represent in matrix format this will be called our vector space model.

C) Similarity Distance:

Clustering requires definition of a distance-measure which assigns a numeric value to the extent of difference between two documents and which the clustering algorithm uses for making different groups of a given dataset. Term variance will get count to compute similarity distance.

D) Estimate Cluster:

It consist of set of data partitions, appropriate partition will get selected by assigning index value to every input which will decide in which cluster it will belong.

E) Clustering Algorithms:

Clustering algorithms applied in this phase which we have previously described in detailed.

F) Search Optimization:

After clustering we are going to add search optimization technique using Pattern Matching algorithm. We can search required file in cluster which may contain thousands of data files.

5. Conclusion

In this thesis we investigated many existing algorithms and proposed two new ones. We conclude that it is hardly possible to get a general algorithm, which can work the best in clustering all types of datasets. Thus we tried to implement two algorithms which can work well in two different types of datasets.

We presented an approach that applies document clustering methods with search optimization. The study of algorithms that eliminate overlapping partitions is worth for achieving goal and it will give labeled cluster as result. This system will help in various fields which is useful to reduce human efforts.

References

[1] Luis Filipe da Cruz Nassif and Edurado Raul Hruschka ”*Document Clustering for forensic analysis: An approach for improving computer inspection*”, IEEE January 2013

[2] S. Decherchi, S. Tacconi, J. Redi, A. Leoncini, F. Sangiacomo, and R. Zunino, *“Text clustering for digital forensics analysis,” Computat. Intell. Security Inf. Syst., vol. 63, pp. 29–36, 2009.*

[3] Chong Su, Qingcai Chen, Xiaolong Wang, Xianjun Men *“Text Clustering Approach Based on Maximal Frequent Term Sets”* Proceedings of the 2009 IEEE International Conference

[4] Michael Steinbach George Karypis Vipin Kumar Department of Computer Science and Egineering, University of Minnesota *“A Comparison of Document Clustering Techniques”*