Vistula University

Faculty: Technology

Program of Study: Computer Science

Kaushal Sureshbhai Dobariya

48176

Document's clustering using hybrid approach.

Bachelor's thesis
written under the supervision of
Dr inz. Marcin Wawryszczuk

Warsaw, 2021

Document's clustering using hybrid approach.

Kaushal Dobariya

Vistula University, Warsaw

Bachelor's in computer science

School of engineering

Email: kaushal10597@gmail.com

ACKNOWLEDGEMENT

The achievement of any venture either huge or little generally relies upon the consolation and rules of numerous others. I accept this open door to offer my thanks to individuals who give me their best bear to making this thesis achievement. I, Kaushal Sureshbhai Dobariya, a student of Bachelor of Computer Science at Vistula University, am extremely grateful to My Guide Dr inz. Marcin Wawryszczuk.

I express my gratitude to Dr inz. Marcin Wawryszczuk. for arranging the dissertation project on a good schedule. Finally, many thanks go to Mr. Edip Senyurek, Head of the program computer engineering, Vistula University, Warsaw who has given her full exertion in controlling all understudies accomplishing the objective just as her consolation to keep up our advancement in track. Without his support and direction, this undertaking would not have emerged.

I might want to value the direction given by another director just as the boards particularly in our undertaking introduction that has improved our introduction abilities by their remark and tips.

Kaushal Sureshbhai Dobariya

ABSTRACT

Author: Kaushal Sureshbhai Dobariya

Title: Documents clustering using hybrid approach

Year: 2021

Language: English

Supervisor: Dr. inz. Marcin Wawryszczuk

Document clustering is one of the best automatic arrangements of document files into clusters so that records inside a bunch have high closeness in a contrast with reports in different groups. It has been concentrated seriously due to its wide materialness in different territories, for example, web mining, web indexes, and data recovery. It measures closeness among reports and gathering comparative archives. Most of the report bunching methods depend on segment grouping and progressive grouping calculation. Among all grouping methods, K-means and Agglomerative bunching procedures are generally utilized for document clustering. The primary point of this thesis work is to improve the nature of archive bunching utilizing the consolidated K-means and hierarchical technique. For the quality proportions of any grouping calculation of different clusters, an assessment matric is utilized. At last, results from the experiment shows that the proposed of a method outperforms K-means algorithm.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

HAC	Hierarchical Agglomerative Clustering
UPGMA	Un-weighted Pair Group Method with Arithmetic Mean
TF	Term Frequency
IDF	Inverse Document Frequency
NLP	Natural Language Processing
PSO	Particle Swarm Optimization
EM	Expectation Maximization
HFTC	Hierarchical Frequent Term based Clustering.
FIHC	Hierarchical Document Clustering Using Frequent Itemset
F2IDC	Fuzzy Frequent Itemset based Document Clustering.
BBC	British Broadcasting Corporation
VSM	Vector Space Model
PHP	Hypertext Preprocessor
IDE	Integrated Development Environment

CHAPTER: 1
INTRODUCTION

The consistent and stunning advancement of computer hardware innovation over the most recent couple of years has prompted huge supplies of ground-breaking and reasonable Hardware, information assortment equipment's, and capacity media. Because of this advancement there is an extraordinary support and inspiration to the information base and data industry to make countless information bases and data archives, which is accessible for exchange the board, data recovery, and information investigation. Hence, innovation headway has given a huge development in the volume of the content archives accessible on the web, advanced libraries and stores, news sources, expansive intranets, and digitized individual data, for example, blog articles and messages. With the expansion in the quantity of electronic records, it is difficult to coordinate, examine and present these archives proficiently by investing manual energy. These have brought difficulties for the successful and proficient association of text records consequently [1].

Data mining is the process of extracting the implicit, beforehand obscure, and conceivably helpful data from information. Record grouping, subset of information bunching, is the procedure of information mining which incorporates ideas from the fields of data recovery, characteristic language preparing, and AI. Record bunching coordinates archives into various gatherings called as groups, where the reports in each bunch share some regular properties as per characterized similitude measure. The quick and excellent report grouping calculations assume a significant function in helping clients to viably explore, sum up, and coordinate the data.

Record bunching includes the utilization of descriptors and descriptor extraction. Descriptors are sets of words that depict the substance inside the bunch. Archive grouping is commonly viewed as a concentrated cycle. Instances of archive grouping incorporate web record bunching for search clients.

Bunching can create either disjoint or covering allotments. In a covering segment, it is feasible for a record to show up in numerous bunches [2] while in disjoint grouping, each archive shows up in precisely one bunch.

Archive Clustering is unique in relation to record order. In record grouping, the classes (and their properties) are known from the earlier, and reports are allotted to these classes, while, in archive bunching, the number, properties, or enrollment (structure) of classes isn't known ahead of time. Subsequently, characterization is an illustration of regulated AI and grouping that of unaided machine learning [2].

Record bunching is partitioned into two significant subcategories, hard grouping, and delicate grouping. Delicate bunching otherwise called covering grouping is again separated into parceling, progressive, and incessant thing set based grouping.

Hard (Disjoint): Hard bunching register the hard task of a record to a group i.e., each archive is allotted to precisely one group; giving a bunch of disjoint groups.

Soft (Overlapping): Soft grouping figure the delicate task i.e., each archive can show up in different bunches; hence, creates a bunch of covering groups. For example, a record talking about Natural language and Information Retrieval will be relegated to Natural language and Information Retrieval bunches.

Partitioning: Partitioning grouping distribute reports into a fixed number of nonempty bunches. The most notable dividing techniques are the K-means and its variations [4]. The essential K-implies technique at first assigns a bunch of objects to a few groups arbitrarily. In every emphasis, the mean of each bunch is determined, and each item is reassigned to the closest mean. It stops when there is no change for any of the groups between progressive emphases.

Hierarchical: Hierarchical report grouping is to fabricate dendrogram, a progressive tree of bunches, whose leaf hubs speak to the subset of a record assortment.

Hierarchical Agglomerative Clustering (HAC) and Un-weighted Pair Group Method with Arithmetic Mean (UPGMA) fall in this classification.

Frequent thing set based: These techniques utilize continuous thing sets created by the affiliation rule mining to group the archives. Likewise, these strategies decrease the dimensionality of term includes productively for enormous datasets, hence improves the exactness and versatility of the bunching calculations. Another bit of leeway of regular thing set based grouping technique is that each bunch can be marked by the acquired successive thing sets shared by the reports in a similar group.

1.1 Problem Definition

In applications like information retrieval system, finding similar documents, recommendation system, etc. are usually examined hundreds of files. Data in those files consists of unstructured text, processing these documents are very difficult. Data mining tools and techniques are useful to process the documents. Selection of appropriate features of the documents, selection of appropriate similarity measure, selection of appropriate clustering method, assessment of the quality of the clusters, Implementation of the clustering algorithm in an efficient way by making optimal use of available memory and CPU resources, Associate meaningful label to each final cluster are important features when performing document clustering. The main issue arise when the large number of documents are used to processes because of its high dimensionality.

1.2 Motivation

Report grouping has been utilized in a wide range of territories of text mining and data recovery. At first it was utilized for improving the exactness and review in data recovery frameworks and finding closest neighbors of a record. Later it has likewise been utilized for getting sorted out the outcomes returned by a web crawler and creating progressive groups of archives.

The main aim of this dissertation work is to improve the effectiveness of existing document clustering algorithms by using hybrid clustering approach.

1.3 Applications

Archive grouping is solo learning and is applied in numerous fields of business and science. At first, archive bunching was read for improving the exactness or review in data recovery frameworks. Archive grouping has additionally been utilized to consequently produce progressive bunches of reports. Following are not many uses of report grouping.

Finding Similar Documents: This component is frequently utilized when the client has spotted one great archive in a query item and needs more-like-this. The intriguing property here is that bunching can find reports that are thoughtfully indistinguishable rather than search-based methodologies that are simply ready to find whether the records share a significant number of similar words.

Organizing Large Document Collections: Document recovery centers around discovering reports applicable to a specific question, yet it neglects to tackle the issue of figuring out countless uncategorized archives. The test here is to arrange these reports in a scientific classification indistinguishable from the one human would make given sufficient opportunity and use it as a perusing interface to the first assortment of records.

Duplicate Content Detection: In numerous applications there is a need to discover copies or close copies in numerous records. Bunching is utilized for literary theft identification, gathering of related reports and to reorder indexed lists rankings (to guarantee higher variety among the highest archives). Note that in such applications the depiction of groups is infrequently required.

Recommendation System: In this application a client is suggested articles dependent on the articles the client has just perused. Bunching of the articles makes it conceivable continuously and improves the quality a great deal.

Search Optimization: Clustering enables a ton in improving the quality and proficiency of web crawlers as the client to question can be first contrasted with the groups as opposed to contrasting it straightforwardly with the archives and the query items can likewise be organized without any problem.

1.4 Objective and Scope

The principal target of this thesis work is to join various leveled calculation with the normal K-Means calculation for improving the nature of the groups. The joined method which is utilized for this work is applied distinctly on the content archives. By applying proper preprocessing steps, this procedure can likewise be utilized to group different kinds of records like pdf, docs, and so forth.

1.5 Structure of Thesis

this thesis is organized into 6 chapters. A detail outline of the thesis concepts of remaining chapters are follows:

Chapter 2: Gives the background of the clustering techniques which is used for the document clustering and the evaluation measures that are applied to compare two methods.

Chapter 3: Contains brief overview of the Proposed method.

Chapter 4: Represents details of the implementation scenario.

Chapter 5: Represents results and comparative analysis of the existing technique and proposed technique.

Chapter 6: Contains the conclusion of the thesis and the future work.

CHAPTER: 2

DOCUMENTS CLUSTERING LITERATURE ANALYSIS

According [1], document clustering is isolated into two significant subcategories, hard clustering, and delicate clustering. Delicate clustering otherwise called covering clustering is again separated into dividing, various leveled, and re
gular thing set-based clustering.

Hard (Disjoint): Hard clustering register the hard task of a document to a group i.e., each document is allocated to precisely one bunch; giving a bunch of disjoint groups.

Soft (Overlapping): Soft clustering register the delicate task i.e., each document is permitted to show up in different bunches; in this manner, creates a bunch of covering groups. For example, a document talking about Natural language and Information Retrieval will be doled out to Natural language and Information Retrieval bunches.

It is imperative to underline that getting from an assortment of documents to a clustering of the assortment, is not simply a solitary activity. It includes numerous stages, which for the most part involve three fundamental stages: highlight extraction and choice, document portrayal, and clustering.

Highlight extraction begins with the parsing of each document to convey a lot of highlights and disallow a summary of pre-determined stop words which are irrelevant from a semantic perspective. By then, specialist highlights are browsed the plan of removed highlights. Highlight determination is a basic pre-handling technique to wipe out noisy highlights. It diminishes the high dimensionality of the component space and gives better data understanding, which subsequently improves the clustering result, productivity, and execution. It is comprehensively utilized in coordinated learning, like content game plan. Thusly, it is critical for improving clustering capability and sufficiency. Normally utilized part decision estimations are term frequency (TF), inverse document frequency (IDF), and their combinations. These are discussed further in a comparative territory.

In the document portrayal stage, each document is addressed by k highlights with the most elevated choice measurement scores as per top-k choice techniques. Document portrayal strategies incorporate twofold (presence or nonappearance of a component in a document), TF (i.e., inside document term frequency), and TF.IDF.

In the last period of document clustering, the objective documents are assembled into bunches based on the chose highlights and their individual qualities in each document by applying clustering calculations.

2.1 Phase of Document Clustering

Following Figure 2.1 showing the phase of document clustering in details respectively [3].

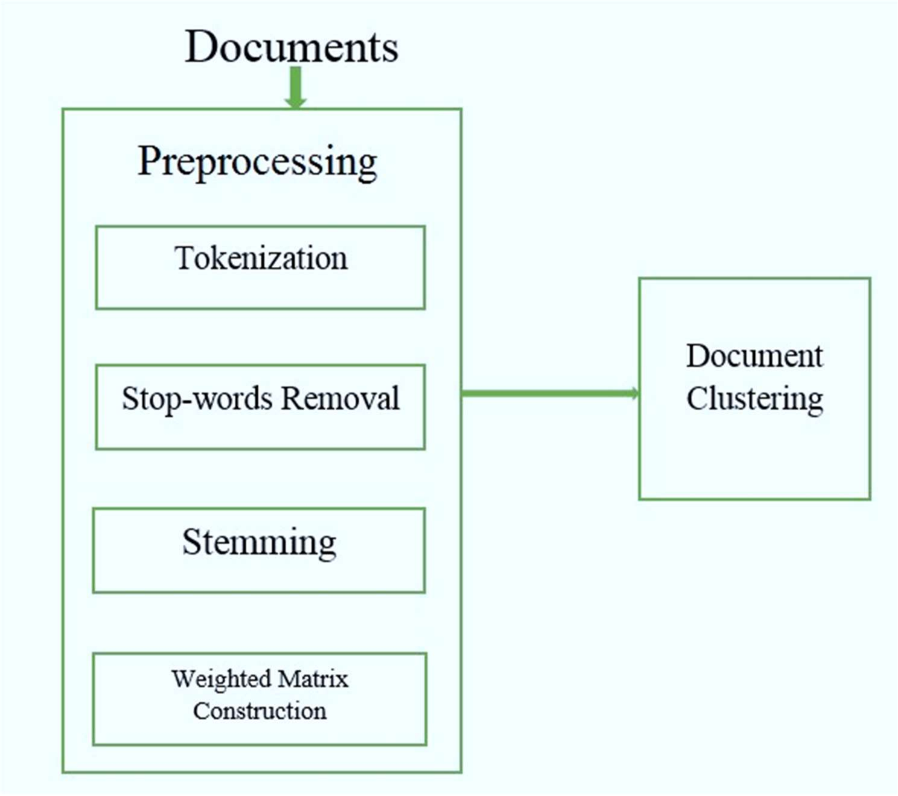

Figure 2.1: Phase of document clustering

Based on this figure we are describing details in below points:

Collection of Data incorporates the cycles like creeping, ordering, sifting, and so on which are utilized to gather the documents that should be grouped, file them to store and recover in a superior manner, and channel them to eliminate the additional data, for instance, stop words.

Tokenization: parts sentences into isolates tokens, run of the mill words. Generally, sophisticate techniques, drawn from the field of NLP, parse the syntactic design of the content to pick huge terms or chunks, for example, thing phrases.

Stemming: The interaction of diminishing words into their base structure and stem. For instance, the words associated", association", associations" are completely diminished to the stem interface." Porter's calculation is the true standard stemming calculation.

Stop-word Removal: A stop-word is characterized as a term, which isn't thought to pass on any significance as a measurement in the vector space (for example not setting). An ordinary technique can eliminate stop-words by contrast each term and an accumulation of known stop-words. Another methodology is to initially apply a grammatical feature tagger and afterward dismissed all tokens that are not things, action words, or descriptors.

Pruning: eliminates words that show up with low frequency all through the corpus. The underlined suspicion in those words, regardless of whether they had any separating power, would frame too little bunches to be valuable. A pre-characterized edge is normally utilized, for example A little or little part of the quantity of words in the corpus. Sometimes words which happen habitually (for example in 45% or more than of the documents) are additionally eliminated.

Post processing: remembers the significant applications for which the document clustering is utilized, for instance, the application that shows the consequences of clustering for prescribing news stories to the clients.

2.2 Term Frequency Inverse Document Frequency (TF.IDF)

In most clustering calculations, the dataset to be bunched is addressed as a bunch of vectors $X=x_1,x_2,,x_n$, where the vector xi is known as the component vector of single article.

In Vector Space Model (VSM), the substance of a document is formalized as a dab in the multidimensional space and addressed by a vector d, for example, $d=w_1,w_2,.....,w_n$, where w_i is the term of weight of the term t_i in one document. The term weight esteem addresses the meaning of this term in a document. To compute the term weight, the event frequency of the term inside a document and in the whole arrangement of documents is thought of. The most broadly utilized weighting plan joins the Term Frequency with Inverse Document Frequency (TF.IDF). The term frequency gives a proportion of the significance of the term inside the specific document. TF.IDF is a factual measure which presents how significant a word is to a document. More regular words in a document are more significant, for example more characteristic of the point.

Let f_{ij}= frequency of term i in document

Now normalize term frequency (tf) across the entire corpus:

$$tf_{ij} = f_{ij}/maxf_{ij} \quad \textbf{(2.1)}$$

The inverse of the document frequencies is as measure of the general importance of the given periods. Terms that show up in a wide range of reports are less characteristic of generally theoretical point.

Let df_i= document frequency of term i = number of documents containing term i

idf_i= inverse document frequency of term i is,

$$log\ 2(N/dfi) \quad \textbf{(2.2)}$$

Where, N: total number of documents. A typical combined term importance indicator is TF.IDF weighting:

$$w_{ij} = tf_{ij} * idf_i = tf_{ij} * log\ 2(N/df_i) \quad \textbf{(2.3)}$$

2.3 Document Clustering Using K-means

K-means is one of the most straightforward unaided learning calculations that tackle the notable clustering issue. The methodology follows a basic and simple approach to group a given data set through a specific number of bunches (expect k bunches) fixed deduced.

The principal thought is to describe k centroids, one for each pack. These centroids should be placed in a shrewdness way by virtue of different region causes assorted result. Thusly, the better choice is to put them whatever amount as could be considered typical far away from each other. The resulting stage is to take each guide having a spot toward a given data set and accomplice it to the nearest centroid. Right when no point is approaching, the underlying advance is done, and an early assembling age is done. Presently we need to re-register k new centroids as focal points of the gatherings coming about in view of the past advance. After we have these k new centroids, another restricting should be done between a comparable data set concentration and the nearest new centroid. A circle has been created. Due to this circle, we may see that the k centroids change their zone step by step until no more changes are done.

In other words, centroids do not move anymore finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

The algorithm is composed of the following steps:

1. Place K concentrations into the space tended to by the things that are being gathered. These centers address starting social affair centroids.

2. Assign each object to the social event that has the closest centroid.

3. When all articles have been distributed, recalculate the spots of the K centroids. Repeat Steps 2 and 3 until the centroids now do not move. This

conveys a division of the articles into packs from which the estimation to be restricted can be determined.

2.4 Document Clustering Using Agglomerative Hierarchical Clustering

Hierarchical techniques produce a settled gathering of portions, with a lone, exhaustive bundle at the top and singleton gatherings of individual concentrations at the base. Each intermediate level can be viewed as joining two bundles from the accompanying lower level (or separating a gathering from the accompanying huger level). The eventual outcome of a hierarchical clustering figuring can be graphically appeared as a tree, called a dendrogram.

This tree graphically shows the solidifying association and the intermediate packs. The dendrogram at the right shows how four centers can be changed over into a singular gathering. For document clustering, this dendrogram gives a logical classification or hierarchical record. Agglomerative techniques are more typical. We summarize the customary agglomerative hierarchical clustering procedure as follows:

- Computer the likeness between all sets of groups, i.e., figure a comparability framework who is worth gives the similitude between the i^{th} and j^{th} bunches.
- Merge the most comparative (nearest) two bunches.
- Update the comparability lattice to mirror the pairwise likeness between the new group and the first bunches.
- Repeat stages 2 and 3 until just a solitary bunch remains.

There are three distinctive agglomerative hierarchical techniques utilized for document clustering.

Intra-Cluster Similarity Technique:

This hierarchical method is taking a gander at the likeness of the entirety of the reports in a cluster to their cluster centroid and is characterized by

$$\boldsymbol{sim}(\boldsymbol{x}) = \sum_{\boldsymbol{deX}} \mathbf{co\,s}(\boldsymbol{d},\boldsymbol{c}) \tag{2.4}$$

Where, d is a document in group, X and c is the centroid of bunch X, i.e., the mean of the document vectors. The 2 decision of which pair of groups to blend is made by determining which pair of bunches will prompt littlest reduction in similitude.

Centroid Similarity Technique: This hierarchical procedure characterizes the likeness of two clusters to be the cosine closeness between the centroids of the two clusters.

UPGMA: This is the UPGMA scheme as described in [2]. It defines the cluster similarity as follows, where d1 and d2 are documents in cluster1 and cluster2, respectively.

$$similarity(cluster1, cluster2) = \frac{\sum_{\cos(d1,d2)}}{size(cluster1) * size(cluster2)} \tag{2.5}$$

2.5 Basic Bisecting K-means Algorithm for finding K clusters.

This assessment begins with a solitary heap of the overall gigantic number of documents and works in the going with way.

- Pick a social occasion to part.
- Find 2 sub-bundles utilizing the crucial K-means assessment. (Bisecting step).
- Repeat stage 2, the bisecting experience, for ITER times and take the split that passes on the clustering with the most fundamental generally speaking closeness.
- Repeat stages 1, 2 and 3 until the ideal number of groups is reached.

There are various approaches to manage pick which group is part. For instance, we can pick the best pack at each development, the one with the most un-in regular closeness, or utilize a reason dependent on both size and by and large likeness. We did various runs and determined that the contrasts between techniques were practically nothing. In the remainder of this paper, we split the best bounty pack.

Note that the bisecting K-means tally can pass on either an un-settled (level) clustering or a hierarchical clustering. For un-settled groups we will often refine the social affairs utilizing the foremost K-means tallies, yet we don't refine the settled packs.

Mindfully speaking, the bisecting K-means check is a badly designed hierarchical clustering figuring, all the while, to maintain a strategic distance from disarray, when we speak of hierarchical clustering calculations, we will mean agglomerative hierarchical assessments of the sort regularly used to gather documents.

At long last, note that bisecting K-means has a period diverse nature which is prompt in the measure of documents. If the measure of groups is huge and tolerating refinement isn't utilized, bisecting K-means is amazingly more proficient than the ordinary K-means tally.

2.6 Expectation Maximization

The EM algorithm fall inside a subcategory of the level clustering algorithms, called Model-based clustering. The model-based clustering expects that data were produced by a model and afterward attempts to recuperate the first model from the data. This model at that point characterizes clusters and the cluster membership of data.

The EM algorithm is a speculation of K-Means algorithm wherein the arrangement of K centroids as the model that produce the data. It switches back and forth between an assumption step, comparing to reassignment, and a boost step, relating to precomputation of the boundaries of the model.

2.7 Frequent item-set-based Document Clustering

These strategies utilize frequent item-sets created by the affiliation rule mining to cluster the documents. Additionally, these strategies diminish the dimensionality of term includes effectively for extremely huge datasets, subsequently improves the exactness and versatility of the clustering algorithms.

Another benefit of frequent-item-set based clustering strategy is that each cluster can be named by the acquired frequent item-sets shared by the documents in a similar cluster [9]. These strategies incorporate Hierarchical Frequent Term based Clustering (HFTC), Hierarchical Document Clustering Using Frequent Itemset (FIHC), and Fuzzy Frequent Item-set-based Document Clustering (F2IDC).

HFTC strategy limits the cover of clusters in terms of shared documents. Be that as it may, the trials of Fung et al. showed that HFTC is not adaptable. For an enormous dataset in [9] FIHC algorithm is given where frequent itemset got from the affiliation rule mining are utilized to develop a hierarchical theme tree for clusters. FIHC utilizes just the worldwide frequent items in document vectors, which lessens the dimensionality of the document set. Accordingly, FIHC is not just adaptable, yet in addition exact [9].

In F2IDC fuzzy affiliation rule mining is joined with WordNet. A term progressive system created from WordNet is applied to find summed up frequent itemset as competitor cluster names for gathering documents. The created clusters with theoretical names are clearer than clusters commented on by disconnected terms for distinguishing the substance of individual clusters.

2.8 Feature based Clustering Approach.

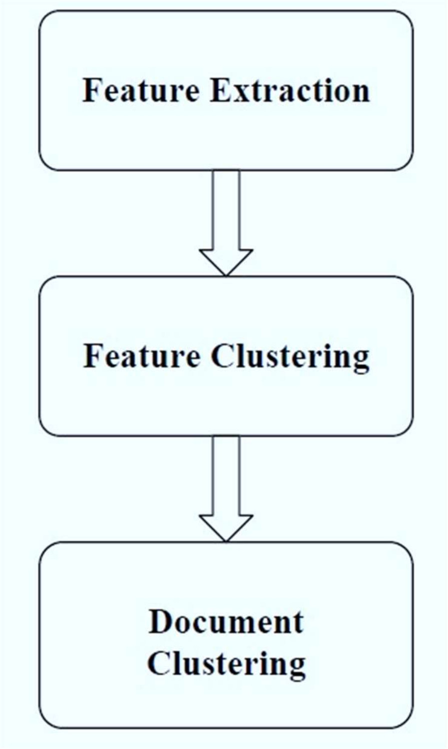

Figure 2.2: Clustering Approach used for Cluster of the Documents.

2.8.1 Feature Extraction

This is utilized for extraction of highlights (significant words and expressions for this situation) from the archives. We have used Named-Entity tagger and frequency of unigrams and bigrams to extract the important words from the documents [1][10].

2.8.2 Feature Clustering

This is the main stage where in the removed highlights are clustered dependent on their co-occurrence. For this we attempted numerous algorithms and discovered Multi-level chart bunching algorithms to be best for enormous informational collection as it lessens the time taken generally.

2.8.3 Document Clustering

This is the last stage wherein reports are clustered utilizing the element clusters. For this we have utilized a straightforward methodology in which a report is allocated to the cluster of expressions of which it has the greatest words.

2.9 Multi-Level Graph Partitioning Algorithm

The multilevel algorithms are graph clustering algorithms which take a graph as input in which an edge defines the similarity between two nodes it is connecting[11]. Based on these similarities it clusters the nodes.

The overview of a multilevel algorithm is this: The three phases are:

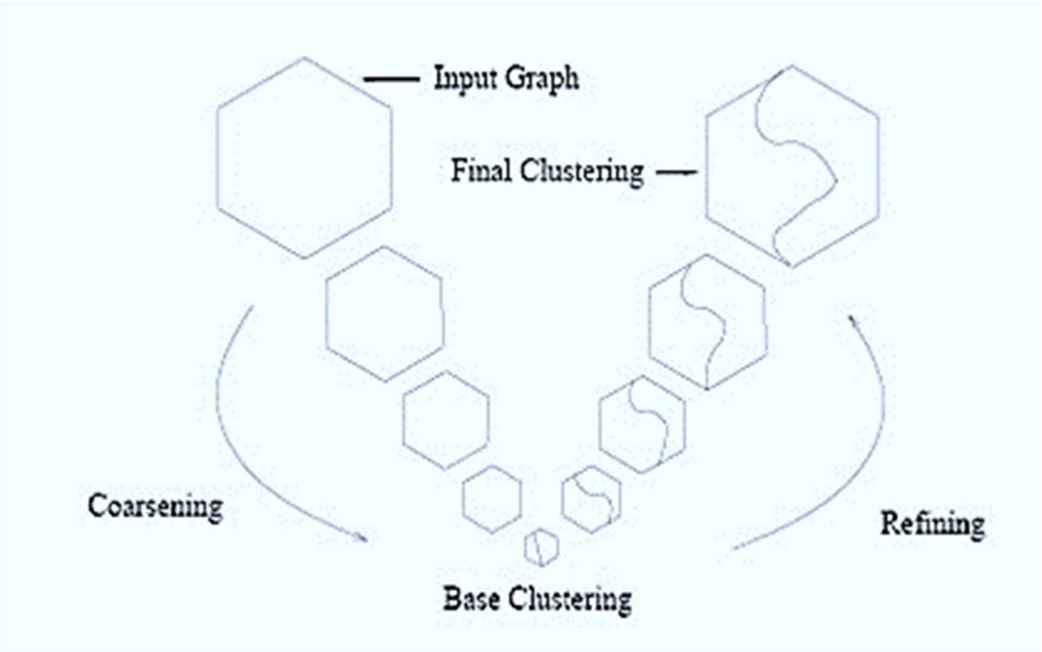

Figure 2.3: Multi-level graph partitioning algorithm

2.9.1 Coarsening

In the coarsening stage the chart is over and over transformed into more modest and more modest diagrams by combining set of nodes to frame supernodes. When combining a bunch of nodes into a solitary supernode, the edge loads out of the supernode are taken to be the amount of the edge loads out of the original nodes.

2.9.2 Base Clustering

The graph is coarsened until it turns out to be sufficiently little to be clustered effectively and adequately. Now base clustering is performed by straightforwardly clustering the coarsened graph. The algorithms utilized for base clustering are the standard graph clustering algorithms.

2.9.3 Refining

In the refinement stage, the clustered starting graph is procured by segregating the hubs which were consolidated in the coarsening stage. In the given graph Gi, the graph Gi-1 is acquired which is the graph utilized in level I-1 of the coarsening stage.

The clustering in Gi actuates a clustering in Gi1 as follows:
If we assuming a supernode of Gi is in a cluster of c, all hubs in Gi-1 framed from that supernode are in cluster c. This yields an underlying clustering for the graph Gi-1, which is improved utilizing a refinement algorithm.

2.10 Hybrid PSO+K-means Algorithm

In the hybrid PSO+K-means algorithm, the multidimensional document vector space is demonstrated as an issue space. Each term in the document dataset addresses one element of the issue space. Each document vector can be addressed as a dab in the issue space. The entire document dataset can be addressed as a different measurement space with numerous dabs in the space. The hybrid PSO+K-means algorithm incorporates two modules, the PSO module, and the K-means module.

At the underlying stage, the PSO module is executed for a brief period to look for the cluster's centroid areas. The areas are moved to the K-means module for refining and creating the last ideal clustering solution[24]

2.11 Similarity Measures for Document Clustering

Cluster analysis techniques depend on estimations of the closeness between a couple of articles. The assurance of likeness between a couple of items includes three significant steps[1][16]:

- The determination of the factors to be utilized to describe the articles.
- The determination of a weighting plan for these factors.
- The determination of a likeness coefficient to decide the level of similarity between the two property vectors.

Exact clustering requires an exact meaning of the closeness between a couple of items, regarding either the pair-wise likeness or distance. An assortment of similitude or distance measures have been proposed and generally applied, like cosine closeness, Jaccard connection coefficient, Euclidean distance, and relative entropy.

2.11.1 Metric

Not every distance measure is a metric. To qualify as a metric, a measure d must satisfy the following four conditions[26].

Let x and y be any two objects in a set and d(x, y) be the distance between x and y.

1. The distance between any two points must be nonnegative, that is, $d(x,y) > 0$.
2. The distance between two objects must be zero if and only if the two objects are identical, that is, $d(x,y) = 0$ if and only if x = y.
3. Distance must be symmetric, that is, distance from x to y is the same as the distance from y to x, ie. $d(x,y) = d(y,x)$.
4. The measure must satisfy the triangle inequality, which is $d(x,z) =d(x,y) +d(y,z)$.

2.11.2 Euclidean Distance

Euclidean distance is a standard metric for geometrical issues. It is the regular distance between two concentrations and can be easily assessed with a ruler in a couple dimensional space. Euclidean distance is extensively used in clustering issues, including clustering text. It satisfies even more than four conditions and likewise is an authentic metric. It is moreover the default distance measure used with the K-means calculation.

Estimating distance between text documents, given two documents da and db addressed by their term vectors ta and tb individually, the Euclidean distance of the two ta and tb documents is characterized as

$$D_E(\underset{ta}{\to}, \underset{tb}{\to}) = \sqrt{\sum_{t=1}^{m} |w_{(t,a)}, w_{(t,b)}|^2} \quad (2.6)$$

where the term set is $T = t_1, t_2, \ldots, t_m$. As mentioned previously, we use the $tf_i df$ value as term weights, that is $w_{(t,a)} = tf_i df(d_a, t)$.

2.11.3 Cosine Similarity

Exactly when reports are tended to as term vectors, the likeness of two records relates to the connection between the vectors. This is estimated as the cosine of the point between vectors, that is, the alleged cosine likeness. Cosine likeness is potentially the most standard closeness gauges applied to message records, for instance, in different information recuperation applications and clustering also. Given two documents ta and tb, their cosine closeness is

$$SIM(\underset{ta}{\to}, \underset{tb}{\to}) = \frac{\underset{ta}{\to}\underset{tb}{\to}}{|\underset{ta}{\to}| X |\underset{tb}{\to}|} \quad (2.7)$$

Where ta and tb are m-dimensional vectors over the term set T = t1,...,tm. Each measurement addresses a term with its weight in the document, which is non-negative. Subsequently, the cosine similarity is non-negative and limited between

[0,1].A huge property of the cosine likeness is its self-rule of document length. For example, consolidating two indistinguishable copies of document d to get another pseudo document d', the cosine equivalence among d and d' is 1, which suggests that these two documents are regarded to be indistinguishable. Meanwhile, given another document l, d and d' will have a similar similitude worth to l, that is, sim(td,tl) = sim(td,tl). At the end of the day, documents with similar synthesis yet various sums will be dealt with indistinguishably. Cautiously talking, this does not satisfy the second condition of estimation, in light of the fact that after all the blend of two copies is a substitute thing from the principal documents. In any case, eventually, when the term vectors are normalized to a unit length like 1, and for the present circumstance, the depiction of d and d0 is something practically the same.

2.11.4 Jaccard Coefficient

The Jaccard coefficient, which is sometimes alluded to as the Tanimoto coefficient, measures comparability as the convergence partitioned by the association of the articles [2]. For text documents, the Jaccard coefficient looks at the entirety weight of shared terms to the aggregate load of terms that are available in both two documents however are not the common terms. The proper definition is:

$$SIM(\underset{ta}{\rightarrow},\underset{tb}{\rightarrow}) = \frac{\underset{ta}{\rightarrow}\underset{tb}{\rightarrow}}{|\underset{ta}{\rightarrow}|+|\underset{tb}{\rightarrow}|-\underset{ta}{\rightarrow}\underset{tb}{\rightarrow}} \qquad (2.8)$$

The Jaccard coefficient is a similitude measure and ranges somewhere in the range of 0 and 1. It is 1 when the ta = tb and 0 when ta and tb are disjoint, where 1 means the two items are something similar and 0 means they are totally extraordinary. The relating distance measure is DJ = 1−SIM and we will utilize DJ rather in resulting tests.

Normally, the cosine work is utilized to gauge the closeness between two documents, yet it may not work well when the clusters are not all around isolated.

2.12 Evaluation of Document Clustering Algorithm

Perhaps the main issues in cluster examination is the assessment of the clustering results. Assessment is the investigation of the yield to see how well it imitates the first construction of the data[1][2][4].

The methods of assessment are partitioned in two sections:

- **Internal quality measure**: Here, the overall comparability measure is used subject to the pair's clever likeness of records, and no external data is used. The cohesiveness of clusters can be used as an extent of cluster closeness. One methodology for calculating the cluster cohesiveness is the utilization of the weighted likeness of the inside cluster closeness.

- **External quality measure:** Some outside data for the data is required. One outside measure is entropy. It gives a proportion of goodness to un-settled clusters or for the clusters at one degree of progressive clustering. Another outer measure is the F-measure which estimates the adequacy of various leveled clustering.

Shannon Entropy: Entropy is utilized as a proportion of the nature of the clusters [2][15]. For each cluster, the classification dispersion of information is determined first for example leave p_{ij} alone the likelihood that an individual from cluster *j* has a place with classification *i*. At that point, the entropy of each cluster j is determined as :

$$E_j = -\sum P_{ij} log\left(\rho_{ij}\right) \tag{2.9}$$

The total entropy is calculated by adding the entropies of each cluster weighted by the size of each cluster: Where m is the all-out number of clusters, n_j is the size of j^{th} cluster and n is the absolute number of documents.

F-measure: This is a total of accuracy and review idea of information retrieval[2][15]. Accuracy is the ratio of the quantity of important documents to the all-out number of documents recovered for a query. Review is the ratio of the quantity of important documents recovered for a query to the absolute number of significant documents in the whole collection. For cluster *j* and class *i*

$$Recall(i,j)=n_{ij}/n_j \qquad (2.10)$$

$$Precision(i,j)=n_{ij}/n_j \qquad (2.11)$$

where n_{ij} is the number of members of class i in cluster j, n_j is the number of members of cluster j and n_i is the number of members of class i.

The F-measure of cluster j and class i is calculated from precision and recall as

$$F(i,j) = \frac{(2 * Recall(i,j) * Precision(i,j))}{(Precision(i,j) + Recall(i,j))} \qquad (2.12)$$

$$F = \sum_i \frac{n_i}{n} max\{F(i,j)\} \qquad (2.13)$$

where the max is taken over all clusters at all levels, and n is the number of documents.

Higher value of F-measure indicates better clustering

CHAPTER: 3

PROPOSED APPROACH

Progressive clustering algorithms assemble a chain of importance of documents. One of the fundamental issues with progressive clustering is that the clustering of documents cannot be changed whenever it is clustered. Accordingly, various leveled clustering endeavors to save the local improvement basis yet not the overall advancement model. We can save the worldwide enhancement model by clustering lost documents.

The proposed algorithm utilizes base up agglomerative various leveled clustering algorithms to address this issue. The client passes the K' cluster information (centroids) registered from the K-means algorithm to the UPGMA algorithm to address the irregularities that happened because of some unacceptable choice made while combining a cluster.

First, apply the K-means algorithm on the assortment of the document for a specific estimation of the K' until K' number of document clusters produced. For comparative documents or most reduced intra-cluster comparability esteem is picked at each progression to blend. The created document clusters ought not be unfilled. At that point, we ascertain the centroids for every one of the subsequent clusters. Every one of these centroids addresses clusters with its documents.

Algorithm:

1. Initially each document is considered as one cluster.
2. Merge two clusters utilizing the K-means algorithm.
3. Repeat Steps 1 and Steps 2 until the K' < K number of clusters are created.
4. Compute the centroids for every K cluster with the end goal that each document in an assortment has a place with one of these centroids.
5. Construct a K' * K' closeness grid between clusters.

6. Merge two clusters with comparative centroid or split clusters of documents in the very cluster that are not comparable.
7. Update the cluster closeness grid.
8. Repeat Steps 6 and Steps 7 until the K clusters are created.

In Steps 1-3, Initially, each document considered as one cluster. Union two documents or two clusters utilizing the K-means algorithm dependent on Euclidean distance until the quantity of clusters K' is created or each document has a place with one of the produced clusters.

In Step 4, figure the centroids for each produced K' clusters.

In Steps 5-8, run the agglomerative various leveled clustering algorithm on the centroids of these document clusters for a given estimation of K (given in the algorithm) to create a bunch of K centroid clusters.

Program Code:

Program code for clustering document from localhost drive (here file will add to c drive in particular software installation folder.)
The code for hybrid documents clustering from the folder.

```
<!doctype html>

<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <meta name="description" content="">
        <title>Stream</title>
        <!-- Bootstrap core CSS -->
        <link href="/stream/assets/bootstrap/bootstrap.min.css"
rel="stylesheet">
    </head>
<body>
<?php
ini_set('display_errors', 1);
error_reporting(E_ALL);
```

```
require_once 'lib/stemmerClass.php';
require_once 'lib/PorterStemmer.php';
$directory = getcwd()."/doc/";
$longlistDir = getcwd()."/doc_array/";

// ****** Step 1 Start*****
?>
<div class="container">
<h1 class='text-center'>Step 1:</h1>
<table class="table table-bordered">
    <thead>
        <tr>
            <th>Document Name</th>
            <th>Content</th>
        </tr>
    </thead>
    <tbody>
    <?php
        $files2 = scandir($directory, 1);
        foreach($files2 as $key=>$file){
            if($file != "." && $file != "..") {
                $myfile = fopen($directory.$file, "r")or die("Unable to
open file!");
                // echo file_get_contents($directory.$file);
    ?>
                <tr>
                    <td><?php echo $file; ?></td>
                    <td><?php echo
file_get_contents($directory.$file); ?></td>
                </tr>
    <?php
                fclose($myfile);
            }
        }
    ?>
    </tbody>
</table>
</div>
<?php
// ****** Step 1 End*****

// ****** Step 2 Start*****
$longlist_files = scandir($longlistDir, 1);
foreach($longlist_files as $key=>$longlist_file){
    if($longlist_file != "." && $longlist_file != "..") {
        $myfile = fopen($longlistDir.$longlist_file, "r")or die("Unable
to open file!");
        while (($line = fgets($myfile)) !== false) {
            $longlist_array[$key] = array_map('strtolower', explode("
", $line));
        }
    }
}
foreach($files2 as $key=>$file){
    if($file != "." && $file != "..") {
        $myfile = fopen($directory.$file, "r")or die("Unable to open
file!");
        $stream_file = "";
        while (($line = fgets($myfile)) !== false) {
            $stream_file = array_map('strtolower', explode(" ",
$line));
```

```
            foreach($longlist_array as $longlist){
                $stream_file = array_diff($stream_file, $longlist);
            }
        }
        $stemmer_obj = new Stemmer();
        $stem_res[$key]['file'] = $file;
        $stem_res[$key]['content'] =
$stemmer_obj->stem_list($stream_file);
        fclose($myfile);
    }
}
?>
<div class="container">
    <h1 class='text-center'>Step 2:</h1>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Document Name</th>
                <th>Content</th>
            </tr>
        </thead>
        <tbody>
        <?php
            foreach($stem_res as $stem_arr){
        ?>
                <tr>
                    <td><?php echo $stem_arr['file']; ?></td>
                    <td>
                        <?php
                            $steam_content = $stem_arr['content'];
                            foreach($steam_content as $stem_val){
                                echo $stem_val." ";
                            }
                        ?>
                    </td>
                </tr>
        <?php
            }
            ?>
    </tbody>
</table>
</div>
<?
// ****** Step 2 End*****

// ****** Step 3 Start*****
?>
<div class="container">
    <h1 class='text-center'>Step 3:</h1>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Document Name</th>
                <th>Content</th>
            </tr>
        </thead>
        <tbody>
        <?php
            $total_array = array();
            $i = 0;
            foreach($stem_res as $stem_arr){
        ?>
            <tr>
                <td><?php echo $stem_arr['file']; ?></td>
                <td>
                    <?php
                        $steam_content = $stem_arr['content'];
                        $group_words = 
array_count_values($steam_content);
                        foreach (array_keys($total_array + 
$group_words) as $key) {
                            $total_array[$key] = 
(isset($total_array[$key]) ? $total_array[$key] : 0) + 
(isset($group_words[$key]) ? $group_words[$key] : 0);
                        }
                        foreach($group_words as $key=>$group_word){
                            echo $key." - ".$group_word."<br>";
                        }
                    ?>
                </td>
            </tr>
        <?php } ?>
        </tbody>
    </table>
</div>
<?php
// ****** Step 3 End*****

// ****** Step 4 Start*****
$max_count = 3;
$match_arr = array();
array_multisort($total_array, SORT_DESC);
$i=0;
$last_val=0;
$result = array();
foreach($total_array as $key=>$array_val){
    if($last_val == 0){
        $last_val = $array_val;
    }
    if(($last_val >= $array_val) && ($i < $max_count)){
        if($last_val == $array_val){
            $match_arr['group_'.$i][$key] = $array_val;
            $result['group_'.$i] = array();
        }else {
            $i = $i+1;
            if($i < $max_count){
                $match_arr['group_'.$i][$key] = $array_val;
                $result['group_'.$i] = array();
            }
        }
    }
    $last_val = $array_val;
}
// print_r($match_arr);
?>
<div class="container">
    <h1 class='text-center'>Step 4:</h1>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Group</th>
                <th>Content</th>
            </tr>
        </thead>
        <tbody>
        <?php
            $i = 1;
            foreach($match_arr as $match_row){
                echo "<tr>";
                echo "<td>Group ".$i."</td>";
                echo "<td>";
                foreach($match_row as $match_word=>$match_key){
                    echo $match_word." - ".$match_key."<br>";
                }
                echo "</td>";
                echo "</tr>";
                $i++;
            }
        ?>
        </tbody>
    </table>
</div>
<?php
// ****** Step 4 End*****

// ****** Step 5 Start*****
$i=0;
$result['no_match'] = array();
foreach($stem_res as $stem_arr){
    $steam_content = $stem_arr['content'];
    $group_words = array_count_values($steam_content);
    array_multisort($group_words, SORT_DESC);
    $no_match = 0;
    foreach($group_words as $group_key=>$group_word){
        foreach($match_arr as $match_key=>$match_inner) {
            if(array_key_exists($group_key, $match_inner)){
                array_push($result[$match_key], $stem_arr['file']);
                $no_match = 1;
                break 2;
            }
        }
    }
    if($no_match == 0){
        array_push($result['no_match'], $stem_arr['file']);
    }
    $i++;
}
// print_r($result);
?>
<div class="container">
    <h1 class='text-center'>Step 5:</h1>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Group</th>
                <th>Document Name</th>
            </tr>
        </thead>
        <tbody>
        <?php
            foreach($result as $result_key=>$result_arr){
                echo "<tr>";
                if($result_key == 'no_match'){
                    echo "<td>No Match</td>";
                }else {
                    echo "<td>";
                    foreach($match_arr[$result_key] as 
$match_key=>$match_row){
                        echo $match_key."<br>";
                    }
                    echo "</td>";
                }
                // echo "<td>".$result_key."</td>";
                echo "<td>";
                foreach($result_arr as $document_name){
                    echo $document_name."<br>";
                }
                echo "</td>";
                echo "</tr>";
        }
        ?>

        </tbody>
    </table>
</div>
<?php
// ****** Step 5 End*****
?>
<script src="/stream/assets/jquery-3.5.1.min.js"></script>
<script 
src="/stream/assets/bootstrap/bootstrap.bundle.min.js"></script>
</body>
</html>
```

CHAPTER: 4

IMPLEMENTATION SCENARIO

4.1Data Collection

The summary of documents used in this paper is shown in Table 4.1. The details of each data set are described here. 20 news groups data in which contains around 20k news articles categorized in 20 different categories. Data sets from BBC news articles contains more than 2225 documents categorized in 5 categories. Data sets re0 and re1 are from Reuters21578 text categorization test collection Distribution 1.0. Some documents generated from the web for the testing purpose.

Data Set	Source	Number of Documents	Number of Classes	Number of words
Re0	Reuters- 21578	1504	13	11465
Re1	Reuters- 21578	1657	25	3758
20 news	20 News-group	1500	100	10832
BBC	BBC News Articles	2225	5	12256

Table 4.1: Summary Description of Document Sets

4.2 Data Preprocessing

Fetch the Content of File: This is the first sub step in the pre-preparing module in which the substance of the information documents is brought for additional handling.

Stop-word Removing: Brought substance of the info record contains a great deal of stop words for example the words which do not have important meaning.

For instance, assume the information document contains a sentence as "Here we are learning java".

In the above sentence, we have words like "here", "we", "are" which are not important for additional preparing for example stemming. So, remove those words from our unique sentence and we just secret key like "java", "learning" to additional progression for example stemming.

Stemming: This are the step where we are bringing the word to its original base form the Consider the same example.

Example- "Here we are learning java". In this sentence, in the wake of eliminating stop words, we get the words "learning", "java" for stemming.

In stemming, we will bring the word “learning” to its base form as “to learn”.

For this, the algorithms ‘Port stemmer’ is used, but problem with this already existing algorithm is it do not return some words to its bas form with correct spelling or with correct meaning.

For Example: The word worried, It returns “to worri”, and not “To study”. And, words like ‘String’, which are already in base form containing ‘-ing’. That after removing ‘-ing’, words become meaningless.

Preparing Vector Space Model :

Set of text documents is represented as Vector Space Model(VSM). VSM can be represented as V=$\{x_1,x_2,,x_n\}$. Each document x_i is represented as a vector which is called the feature vector. A vector x can be represented as, x= $\{w_1,w_2,,w_n\}$. Where w_i represents the term weight. The term weight can be calculated using TF-IDF (term frequency-inverse document frequency) scheme. Weight of i in document j can be calculated as:

$$W_{ji} = tf_{ji} * idf_{ji} = tf * \log2(n/df_{ji}) \quad (4.1)$$

Where tf_{ji} is the number of times term i has appeared in document j. df_{ji} represents the no of documents in which term i has appeared and n is the total no of documents in collection. This Vector Space Model can also be seen as term document matrix of t*d where t is total no of terms and d is no of documents.

4.3 Implementation of Hybrid Approach

Implementation of hybrid algorithm is done in PHP (Hypertext Preprocess) version 5.4.3 and NuSphere PhpED is used as IDE. A proposed algorithm accepts text files as a input. A Proposed Algorithm is combination of the K-means and hierarchical algorithm. First K-means algorithm is applied on the input data. The output of the K-Means algorithm is the number of clusters therefore each data points belongs to one of the clusters. Then Hierarchical algorithm is Applied on generated cluster. The output of the proposed algorithm is the number of cluster and each cluster contains at-least one data points and the data points are in same cluster are like each other. Finally, Comparative analysis between hybrid approach and regular K-means algorithm based on two parameters, Entropy and F-measures which are used to validate the cluster quality has been done.

CHAPTER: 5

RESULTS AND ANALYSIS

A bunch of documents utilized for assessment has the following features:

1. Number of documents per category.
2. Evenness in various documents in every category.
3. Size of each document for example the quantity of words in each document.
4. Similarity of documents of a similar category contrasted with the likeness of documents of different classes.
5. Number of unique words altogether the documents.

The nature of the aftereffects of the clustering algorithms relies particularly upon the features of the arrangement of documents on which it is applied. For instance, a few algorithms may give great outcomes on account of enormous documents when contrasted with little documents. There are two parameters are utilized to measure the nature of the clustering algorithm.

Entropy: Entropy is utilized as a measure of the nature of the clusters. For each cluster, the category dispersion of data is calculated first for example leave p*ij* alone the likelihood that an individual from cluster *j* has a place with category *i*. At that point the entropy of each cluster *j* is calculated as:

$$E_j = -^{X} p_{ij} \log(p_{ij}) \quad (5.1)$$

The method for total entropy is calculated by adding entropies of each cluster weighted by the size of every single cluster:

$$E_{en} = \sum_{i=1}^{m} ((n_j * E_j)/n \;) \quad (5.2)$$

Where m is the total number of clusters, nj is the size of j^{th} cluster and n are the total number of the documents. The entropy result of the regular K-means and the Hybrid K-means Algorithm is shows in Table 5.1.

Data Sets	Hybrid K-means	K-means
Re0	1.2677	1.3716
Re1	1.4806	1.5241
20 News Group	1.1512	1.3395
BBC News Articles	1.3046	1.2877

Table 5.1: Entropy measured for Hybrid K-means and K-means algorithms.

In view of the above outcomes, the Hybrid K-means algorithm is more exact than the regular K-Means algorithm.

F-Measure: This is a collection of the accuracy and review idea of information recovery. Accuracy is the proportion of the number of pertinent documents to the all-out number of documents recovered for a question. The review is the proportion of the number of pertinent documents recovered for a question to the complete number of important documents in the whole assortment.

The F-Measure consequence of the regular K-means and the Hybrid K-means Algorithm appears in Table 5.2.

Data Sets	Hybrid K-means	K-means
Re0	0.3823	0.3225
Re1	0.3515	0.3369
20 News Group	0.3468	0.3993
BBC News Articles	0.3961	0.4225

Table 5.2: F-Measure measured for Hybrid K-means and K-means algorithms.

In view of the above test results, it is examined that the hybrid methodology gives a superior quality cluster in a contrast with K-means.

The time intricacy of the hybrid methodology is more than the regular K-means. There is no compelling reason to give various clusters by clients in a hybrid methodology.

CHAPTER: 6
CONCLUSION AND FUTURE EXTENSION

It tends to be close from this exposition work that the data mining techniques are more suitable for document clustering. Mistakes in preprocessing steps can decade the presentation of the clustering algorithm.

Hybrid methodology outflanks in a contrast with K-means Algorithm; however, the time intricacy is expanded.

When clustering the enormous number of documents, will produce a huge number of measurements or properties. Accordingly, the time intricacy is expanded, so forward thought is to utilize the dimensionality decrease method prior to applying any clustering algorithm to improve the time intricacy.

Category-based assessments require marking a whole document assortment with different classifications. At the point when countless classifications exist for a broadly useful knowledge archive like Wikipedia or the Internet, this turns into an overwhelming issue. There have been reports of marking document assortments causing "terrible memories" for assessors even though the document assortment comprised of not exactly 1,000,000 documents, had many classifications, had an instrument to help automate classification dependent on rules, what's more, was naming short news wire articles. The issue of enormous scope appraisal has effectively been tended to through the utilization of pooling in specially appointed information recovery assessment. Furthermore, inquiries utilized for assessment are often specific. furthermore, very much defined, unlike a portion of the expansive, lofty classes utilized for document arrangement.

This thesis features that the cluster hypothesis holds with extremely fine-grained document clusters when they are utilized for specially appointed recovery. This was further reinforced by the investigation of the geography of document marks, which shown that lone the nearby neighborhood of the portrayal permits differentiation of importance. Therefore, making document clusters too enormous is likely to contain numerous equidistant documents. It was shown that fine-grained clusters are more effective for cluster-based hunt.

The tale assessment of document clustering utilizing impromptu significance decisions delivered a few findings. The utilization of impromptu importance decisions beats the shortfalls of utilizing classifications. Category-based assessment of document clustering just has use for classification. In any case, impromptu recovery-based assessment has a utilization case for cluster-based recovery. We recommend that assessing document clustering in the circumstance of its utilization is more solid than a classification-based assessment. If you need to do classification, fabricate a classifier. Document clustering is driven by setting comparable things together. Since two documents exist in a category doesn't really mean they are comparable. The cluster hypothesis is straightforwardly determined by the closeness between documents, where documents that are like one another, are likely to be pertinent to a similar information need.

This thesis explores the Divergence from a Random Baseline approach to the assessment of clustering. It considers the differentiation of ineffective clustering’s that perform no useful learning regarding a given measure of cluster quality. This took into consideration the identification of ineffective clustering’s in the Mining track.

REFERENCES

[1] Neepa Shah, Sunita Mahajan:”Document Clustering: A Detailed Review”, In_ternational Journal of Applied Information Systems, Volume 4, Issue 5, October 2012.

[2] Michael Steinbach, George Karypis, Vipin Kumar: ”A Comparison of Document Clustering Techniques”, In KDD Workshop on Text Mining, 2000.

[3] Sanjivani Tushar Deokar: ”Text Documents clustering using K Means Algorithm”, International Journal of Technology and Engineering Science, Volume 1, Issue 4, July 2013.

[4] Chandan Jadon, Ajay Khunteta: ”A New Approach of Document Clustering”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013.

[5] Bhagyashree Umale, Nilav M: ”Survey on Document Clustering Approach for Forensic Analysis”, International Journal of Computer Science and Information Technology, Volume 5, Issue 3, April 2014.

[6] Francis Musembi Kwale: ”A Critical Review of K-Means Text Clustering Algo_rithms”, International Journal of Advanced Research in Computer Science, Vol_ume 4, Issue 9, July-August 2013.

[7] Manjot Kaur, Navjot Kaur: ”Web Document Clustering pproaches Using K_Means Algorithm”, International Journal of Advanced Research in Computer Sci_ence and Software Engineering, Volume 3, Issue 5, 2013.

[8] Omar Kettani, Faycal Ramdani, Benaissa Tadili: ”An Agglomerative Clustering Method for Large Data Sets”, International Journal of Computer Applications, Volume 92, Issue 14, April 2014.

[9] Rekha Baghel, Dr. Renu Dhir: ”A Frequent Concepts Based Document Clustering Algorithm”, International Journal of Computer Applications, Volume 4, Issue 5, July 2010.

[10] Mamta Mahilane, Mr. K. L. Sinha: ”A Survey Paper on Different Techniques of Document Clustering”, Technical Research Organization India, Volume 2, Issue 2, 2015.

[11] Ms.J.Sathya Priya, Ms.S.Priyadharshini: ”Clustering Technique in Data Mining for Text Documents”, International Journal of Computer Science and Information Technology, Volume 3, Issue 1, 2012.

[12] Khaleb B. Shaban: ”A Semantic Approach for Document Clustering”, Journal of Software, Volume 4, Issue 5, July 2009.

[13] Yong Wang and Julia Hodges: ”Document Clustering with Semantic Analysis”, In Proc. of the 39th Annual Hawaii International Conference on System Sciences, Volume 3, 2006.

[14] Peter Willett: ”Recent Trends In Hierarchic Document Clustering: A Critical Review”, Information Processing and Management, Volume 24, Issue 5, 1988.

[15] Ying Zhao and George Karypis: ”Evaluation of Hierarchical Clustering Algorithms for Document Datasets”, Technical Report, June 2002.

[16] Ying Zhao and George Karypis: ”Evaluation of Hierarchical Clustering Algorithms for Document Datasets”, Technical Report, June 2002.

[17] Y. LI, and S.M. Chung: ”Text Document Clustering Based on Frequent Word Sequences”, In Proceedings of the, CIKM, 2005. Bremen, Germany, October 31-November 5.

[18] V. Mary Amala Bai, Dr. D. Manimegalai: ”An Analysis of Document Clustering Algorithms”, IEEE 2010.

[19] A. K. Jain, M. N. Murty, and P. J. Flynn: ”Data Clustering: A Review”, ACM Computing Survey, Volume 31, Issue 3, 1999.

[20] D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Tukey: Scatter/gather: A cluster-based approach to browsing large document collections”, In Proceedings of the ACM SIGIR, 1992.

[21] Y. Zhao and G. Karypis: ”Empirical and theoretical comparisons of selected cri_terion functions for document clustering”, Machine Learning, Volume 55, Issue 3, 2004.

[22] Bishnu Prasad Gautam, Dipesh Shrestha, Members IAENG: Document Cluster_ing Through Non-Negative Matrix Factorization: A Case Study of Hadoop for Computational Time Reduction of Large Scale Documents”, Proceedings of the International Multi Conference of Engineers and Computer Scientist, Volume 1, 2010.

[23] Porter, MF : ”An algorithm for suffix stripping”, Program, Volume 14, Issue 3, pages 130-137, 1980.
http://tartarus.org/ martin/PorterStemmer/def.txt

[24] Xiaohui Cui, Thomas E. Potok: ”Document Clustering Analysis Based on Hy_brid PSO+K-means Algorithm”, Applied Software Engineering Research Group, Computational Sciences and Engineering Division, Oak Ridge National Labora_tory, Oak Ridge, 2005.

[25] B.Drakshayani, E V Prasad: ”Text Document Clustering based on Semantics”, International Journal of Computer Applications, Volume 45, Issue 4, May 2012.

[26] Anna Huang: ”Similarity Measures for Text Document Clustering”, In Proc. of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC), pp. 4956, April 2008.

[27] K.P.N.V.Satyasree, Dr.J V R Murthy: ”Clustering Based on Cosine Similarity Measure”, International Journal of Engineering Science & Advanced Technology, Volume 2, Issue 3, May-Jume 2012.