A COMBINATORIAL TWEET CLUSTERING METHODOLOGY UTILIZING INTER AND INTRA COSINE SIMILARITY

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Science

in

Software Systems Engineering

University of Regina

by

Navneet Kaur

Regina, Saskatchewan

July, 2015

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Navneet Kaur, candidate for the degree of Master of Applied Science in Software Systems Engineering**,** has presented a thesis titled, ***A Combinatorial Tweet Clustering Methodology Utilizing Inter and Intra Cosine Similarity,*** in an oral examination held on July 13, 2015. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr. Paul Laforge, Electronic Systems Engineering
Supervisor:	Dr. Craig Gelowitz, Software Systems Engineering
Committee Member:	Dr. Luigi Benedicenti, Software Systems Engineering
Committee Member:	Dr. Mohamed El-Darieby, Software Systems Engineering
Chair of Defense:	Dr. Amy Zarzeczny, Johnson-Shoyama Graduate School

ABSTRACT

Data mining techniques are well known and are often used to analyze and explore datasets for meaningful information. Social media, such as Twitter, has emerged as a source of data where millions of tweets are generated everyday. They include tweets from individuals who share thoughts, commentary and their feelings about a wide variety of subjects. Social media also attracts marketers and businesses for the purpose of advertising, brand imaging and getting feedback from users.

Twitter's significant popularity and mass usage has resulted in a very large dataset where virtually any subject that is queried from the Twitter API may return a vast number of tweets. As a result, these tweets can be related to several distinctly different categories. Data mining a large amount of tweets to classify them into meaningful categories is a challenging task because of the often informal language used, the inclusion of URL links, spam and other irrelevant information.

This thesis presents a combinatorial hierarchical clustering methodology that categorizes tweets into meaningful clusters by utilizing inter and intra cluster cosine similarity. Cosine similarity is the degree of relativity between two vectors. This thesis proposes a "Combinatorial Hierarchical Clustering Methodology" as a combination of both agglomerative (Bottom-Up) and divisive (Top-Down) hierarchical clustering approaches that attempts to maximize the clustering effectiveness and quality. The proposed methodology sub-categorizes, divides and combines clusters through an iterative process to help make sorted categories more meaningful. In addition, this

approach does not require a-priori information about the numbers of clusters to be formed but rather forms clusters dynamically based on their determined similarity.

ACKNOWLEDGEMENTS

I would like to express sincere thanks to my supervisor, Dr. Craig M. Gelowitz, for making this research possible. His invaluable guidance, encouragement and support throughout the research are appreciable. I would like to thank the Stem Cell Research Network for funding parts of this research. I would like to acknowledge Faculty of Graduate Studies and Research for funding I received during my study. I would like to thank all my friends and family members who supported me emotionally and financially through every phase of life.

DEDICATION

To,

Mom, Dad and Harman

whose love and support have made this research possible

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

Social media has become a significant data-mining source for researchers to analyze and extract meaningful information for a variety of purposes. Some examples of current research areas in social media data mining include the analysis of:

- Trending topics in social media.
- Methodologies to detect spam.
- Sentiment of users on specific topics.
- Semantics of posted data.
- The visualization of dynamic trends within social media data.

Social networking services such as Facebook and Twitter enable users to share information regardless of geographical location. This has resulted in collection of a large volume of data that causes computational challenges [52]. Twitter has been chosen as the source of data for this research.

1.1. Twitter

Twitter's own statistics show that there are currently 284 million monthly active users and about 500 million tweets being sent everyday [3]. Twitter is a micro-blogging social networking service where individuals can communicate and share their opinions via a maximum of 140 character messages, also known as tweets. Twitter has emerged as a social media networking platform for not only the average individual user, but also for celebrities and businesses. As such, it has emerged also as a powerful marketing tool that organizations use for brand imaging and for obtaining feedback from customers.

Consequently, many organizations have invested resources in Twitter for the following reasons:

- To promote their products and services to attract more customers.
- As a cheap source of advertisement and promotion.
- To receive feedback from their customers.
- To help determine their market position.
- To analyze their competitors and other potential obstacles that may affect their success.

1.2. Research Motivations

In this thesis, the intention is to take un-labeled tweets and then group them together into appropriate categories so that similar tweets are sorted into appropriate categories. The motivation for this research is to provide contributions in the methodologies for clustering large numbers of tweets into meaningful categories.

Clustering is different from classification because it is applied on un-labeled data to form groups and only then used to classify that data. Clustering is also known as an un-supervised machine learning technique.

Data mining tweets and grouping them into meaningful categories is a challenging task because Tweets:

- Have a limited number of words due to the limited number of characters allowed by Twitter.
- Often contain informal language.

- Include URL links.
- May be spam. It has been estimated that 40 % of accounts on social media sites are used for spam and 8 % of messages exchanged are spam [47].
- May contain irrelevant information.

1.3. Hypothesis

The proposed methodology will preprocess tweets to remove the clutter; extract relevant features from each tweet, and cluster the data into meaningful categories.

The similarity of tweets in this thesis is to be measured using cosine similarity. Cosine similarity analyzes the degree of relativity between two n-dimensional vectors based on tweet content. It treats two documents (or tweets) as unit vectors and calculates how far they are in terms of angular distance.

Cosine similarity is used in this work because it does not consider the magnitude of the document vector. Therefore, the length of the document does not influence similarity score. Only the feature weights in the document vector are counted to calculate the similarity [58].

For example,

a. If cosine similarity is 1, the two vectors have same orientation.
b. If cosine similarity is 0, the two vectors are 90 degree apart.
c. If cosine similarity is -1, the two vectors have exactly opposite orientation.

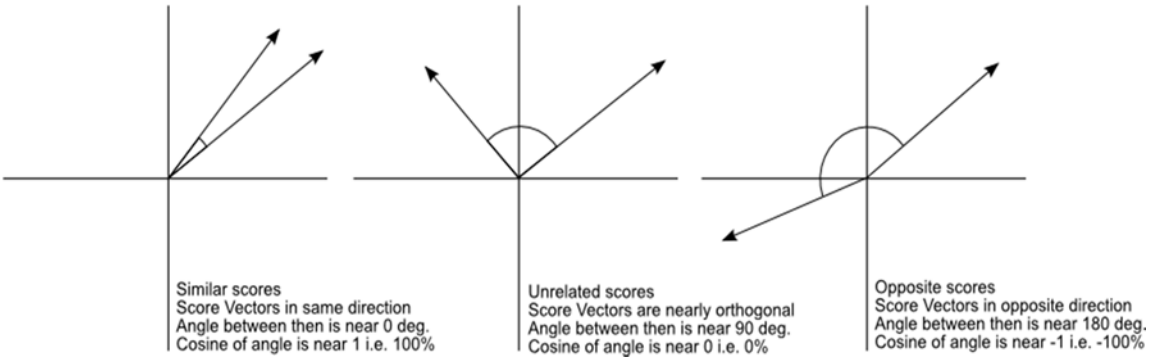


Figure 1.1: Cosine Similarity [48]

Clustering is a process of division of data into groups of similar objects called clusters. A cluster consists of similar objects that are dissimilar to objects in other group. A general definition of clustering is

"*Given a number of objects or individuals, each of which is described by a set of numerical measures, devise a classification scheme for grouping the objects into a number of classes such that objects within classes are similar in some respect and unlike those from other classes. The number of classes and the characteristics of each class area to be determined*" [54].

There are generally two types of clustering algorithms:

a. Partition Based Clustering Algorithm
b. Hierarchical Clustering Algorithm

Partition based clustering algorithms take a number of clusters as input with input parameters from users with the aim of minimizing some objective function. Whereas, hierarchical clustering algorithms group documents together in the form of a tree based on some input parameters using either of two approaches:

- Agglomerative (Bottom-Up)
- Divisive (Top-Down)

The agglomerative approach starts from the bottom of the tree treating each document (or tweet) as individual leaves. It then moves upwards in the hierarchy by grouping the most similar leaves together until root of the tree is reached.

The divisive approach treats all of the documents as one category and divides them into different sub-categories for each hierarchy level. This division is based upon some similarity score until a condition is met.

One of the issues with these approaches is the requirement for a user to provide some input parameters before the clustering process starts. The input parameters affect the formation of the clusters. Some techniques select random candidates that cause the cluster output to vary in successive trials using the same data.

In this work, both of the approaches will be utilized in order to increase the effectiveness and quality of the clustering process. The Bottom-Up approach helps to remove the redundancy in clusters by merging the similar clusters together.

1.3. Research Objectives

The objective of this research is to develop a methodology that does not require a-priori information about the number of clusters or require any other input parameters but rather dynamically forms clusters based on their determined similarity.

The proposed methodology has three primary objectives:

a. Automatically create broad categories based on the appearance of nouns.
b. Calculate the intra-cluster cosine similarity and divide the clusters based on a threshold (Top-Down) to generate multi-level sub-categories based on the broad ones created in (a).
c. Calculate the inter-cluster cosine similarity to merge the resulting sub-categories based on similarity score (Bottom-Up).

1.4. Thesis Contributions

The research contributions of this thesis are:

- A clustering methodology that creates categories of tweets dynamically and does not require user input with respect to initial parameters.
- A clustering methodology that takes categories and divides them incrementally into sub-categories to provide improved clustering and determination of the most similar tweets in each category.
- A combined agglomerative and divisive hierarchical approach to increase the clustering quality and effectiveness.

1.5. Thesis Organization

This thesis has been organized into five chapters:

Chapter 1 includes the introduction to the research, the research motivation, a hypothesis, the research objectives, and the contributions of this research.

Chapter 2 provides a literature review with respect to document clustering. It

reviews several algorithms, approaches and methodologies that have been developed. This chapter indicates how the literature has contributed in this area and what approaches have been used.

Chapter 3 explains the proposed methodology and experimental setup. It provides a theoretical overview of the methodology and explains how this methodology has been applied.

Chapter 4 contains the experiments conducted and their corresponding results. It compares the proposed methodology's output with some other existing clustering approaches presented in chapter 2.

Chapter 5 provides the conclusions and the future work.

2. RELATED WORK

Collecting tweets and analyzing them to retrieve meaningful knowledge patterns and trends has been a recent study area for data mining research. Social media, such as Twitter and Facebook enable individuals to share their personal opinions about a variety of subjects and enables organizations to advertise their products and services. As such, public opinion through this medium can serve as feedback for those organizations. The Search Engine journal analyzed some facts about social media, and wrote that: "93% of marketers are using Social Media" [1].

Allie B. and Merve O. write about social media usage that "Businesses are reaching out to their customers to seek their attention from them" [2]. They also write that Facebook has approximately 1.4 million business pages according to the New York Times in 2010. In addition, companies collect information about what kind of people are showing interest [2].

Unlike Facebook, Twitter is a micro-blogging social networking site that has also captured the attention of marketers. Twitter is a social media service where people communicate and voice opinions via a maximum of 140 character messages called tweets. Celebrities have created twitter accounts that are read by millions of followers every day. Twitter also enables businesses to post advertisements on Twitter accounts to attract more customers. Twitter provides valuable information to marketers such as what kind of customers are visiting their sites and what their interests and feedback are [2]. Twitter's own statistics show that there are currently 284 million monthly active users and about 500 million tweets being sent everyday [3]

2.1. Clustering

Categorizing tweets can be a very tedious and challenging task because tweets often include informal language, the inclusion of URL links, SPAM and other irrelevant information. Various automated and semi-automated systems have been built to group tweets together utilizing text clustering or document clustering techniques.

Text Clustering is defined as an un-supervised machine learning technique that enables the grouping of similar text documents into similar categories. Un-supervised clustering is performed on un-labeled data that has not been classified. This is in contrast to the training of a classifier as in supervised learning techniques.

Char C. Aggarwal and Cheng Xiang Zhai describe the importance of text clustering in their survey of text clustering algorithms. They assert that organized documents make information retrieval and the text browsing process easier and more efficient. They also indicate that text clustering can be applied to a number of tasks such as document organization and browsing (hierarchical distribution of documents), corpus summarization (summary of the whole corpus) and document classification (clustering or grouping similar documents). Their work provides a detailed summary of text clustering techniques, and the recent advancements in social network text clustering [4].

2.2. Text Document Similarity Measuring Techniques

Similarity measure defines the distance between two text documents. Clustering algorithms use some similarity measuring function as a criterion to extract clusters from

large data sets. These functions help in determining how similar or dis-similar two documents are.

Documents are first required to be converted to a vector form because clustering algorithms and distance measuring techniques are unable to interpret documents in their original form. The Vector Space Model (VSM) is a widely used text document representation method.

"In VSM method, a document D_j is represented as a vector of weights of *n-features* extracted from the document:" [51]

$$D_j = (w_{1j,} w_{2j,\ldots,} w_{nj,}) \qquad (1)$$

where n represents the number of features and w_n represents the weight of n^{th} feature.

The weight of a feature defines its contribution to the semantics of document D_j. The weighting term can vary in terms of what the feature is (word, term or n-gram) and the calculation of the weight of a feature. In VSM, a corpus with j documents is represented by j x n matrix, which is known as term-document matrix [51]. Some important similarity measures used in document clustering are explained briefly below:

2.2.1. Euclidean distance

Euclidean distance is defined as sum of square of difference between coordinates of two objects [36, 37]. Therefore, Euclidean distance d between two n-dimensional vectors x and y is calculated as:

$$d = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \qquad (2)$$

For example, Euclidean distance between two document vectors A and B is shown in figure 2.1 below:

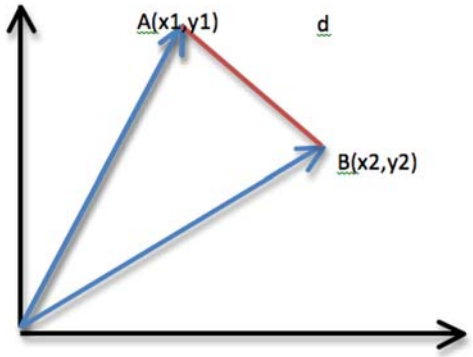


Figure 2.1: Euclidean distance

where d can be calculated as:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (3)$$

2.2.2. Cosine distance

Cosine distance measures the cosine of angle between two document vectors A and B [36,37]. It is calculated as:

$$\cos\theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| * |\vec{B}|} \qquad (4)$$

Figure 2.2 shows the cosine distance between two document vectors A and B.

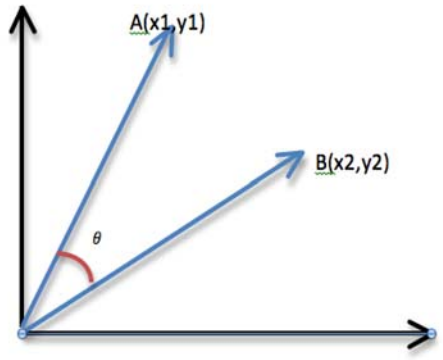


Figure 2.2: Cosine distance

2.2.3. Jaccard Coefficient

The *"Jaccard coefficient measures the distance as the intersection divided by union of objects"* [38,39]. It calculates the sum of the weight of the shared terms and compares it to the sum of the weights of terms occurring in any of the two documents but not shared [38,39].

$$\text{jaccard}(A, B) = \frac{A\ B}{|A|^2 + |B|^2 - A\ B} \qquad (5)$$

or

$$\text{jaccard}(A, B) = |A \cap B| \; / \; |A \cup B| \qquad (6)$$

This is also known as the Tanimoto Coefficient.

2.2.4. Pearson Correlation

This correlation coefficient measures how precisely two sets of data fit on straight line. Its value can range between -1 and 1.

- If a correlation coefficient is 1, it indicates the two documents have a positive linear relationship.
- If a correlation coefficient is -1, it indicates the two documents have a negative linear relationship.
- If a correlation coefficient is 0, it indicates the two documents have no linear relationship [38,40].

The Pearson Correlation $Sim(x, y)$ between two documents x and y is calculated the using formula:

$$Sim(x, y) = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{(\sum x^2 - \frac{(\sum x)^2}{n})(\sum y^2 - \frac{(\sum y)^2}{n})}} \qquad (7)$$

2.2.5. Manhattan distance

It calculates the absolute difference between any two data points and is defined as [41,42]:

$$D_{ij} = \sum_{n=1}^{d} |x_{in} - x_{jn}| \qquad (8)$$

Clustering uses similarity-measuring techniques to calculate the similarity between documents and group them into appropriate categories. Clustering techniques are explained in the following sections.

2.3. Hierarchical Clustering

One of the most popular clustering approaches is Hierarchical Clustering that are generally of two types:

- Agglomerative Hierarchical Clustering (Bottom-Up)
- Divisive Hierarchical Clustering (Top-Down)

Hierarchical methods treat documents as tree like structures called Dendograms [8,9,10].

2.3.1. Agglomerative or Bottom-Up Clustering

Agglomerative clustering starts from the end nodes or leaves considering them as individual clusters and move up in the hierarchy by merging the most similar documents until a final cluster or root node is reached [8,9,10].

2.3.2. Divisive or Top-down Clustering Method

Divisive clustering is employed when there is a one set of documents (a single cluster) that is divided into smaller clusters. Clusters are divided hierarchically at each step so as each cluster represents a unique set of information [8,9,10].

Figure 2.3 shows a dendogram with hierarchical clustering.

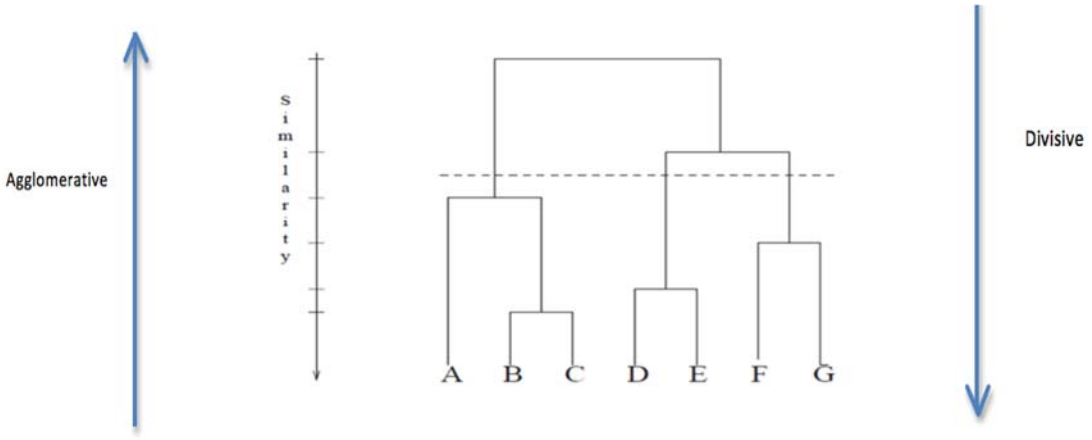


Figure 2.3: A dendogram showing Hierarchical Clustering [7]

It is the merging and splitting operations that are based on similarity measuring techniques.

2.4. Partitioning Clustering

Partitioning clustering algorithms iteratively separate data into different groups or clusters by minimizing some objective function. It is also known as centroid based clustering. In order to partition a data set containing document vectors, a document is compared against the mean of the documents or some nearest neighbor so that the objective function is at a minimum for the document to be associated with that cluster.

2.4.1. k-means

One of the most popular algorithms that fall under this category is the k-means and its variants [11,12,13]. The k –means algorithm consists of four main steps:

1. Choose random k- centroids and calculate the average distance between the document and the centroids.
2. Assign documents to the nearest centroid.
3. Choose new centroids by calculating the mean of all documents in a particular cluster C.
4. Iterate through steps 2 and 3, until the difference between the old and new centroid is less than a threshold [14].

The objective minimization function used in k-means is:

$$\sum_{i=0}^{n} \min(\|x_j - \mu_i\|^2), \mu_j \epsilon\, C \qquad (9)$$

where μ_i is the mean of documents (centroid) in the cluster and x_j is the document [14].

The output of the k-means algorithm is flat and un-hierarchical. The performance of k-means degrades with an increase in the size of the input dataset. The primary limitation of this algorithm is that it expects the clusters to be of similar size in order to make the assignment to a centroid correct [53].

For example, figure 2.4 shows the k-means output on the well-known *Iris Flower Data Set* with k=3. The results show that the k-means fails to separate the three Iris species shown in figure 2.5. It separates one of the clusters into two equal parts. The results may be better with k=2 [53].

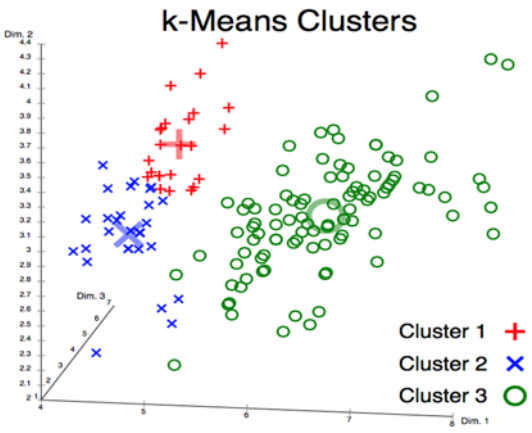


Figure 2.4: k-means output on Iris dataset [53]

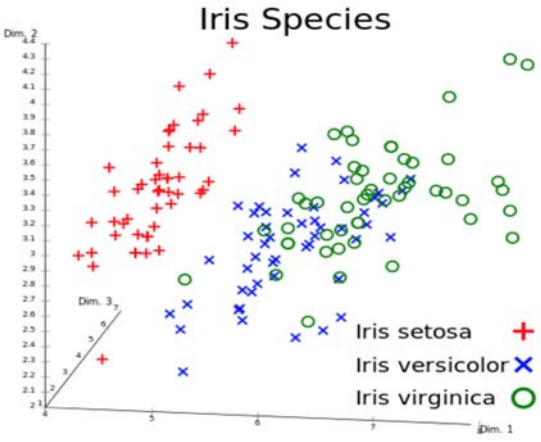


Figure 2.5: Three Iris Species in Iris data set [53]

2.4.2. Bisecting k-means

Another popular variant of the k-means algorithm is the bisecting k-means algorithm. It produces results in the following manner:

1. Choose a cluster to split.
2. Run a generic k-means algorithm to generate two sub-clusters. This step is called the bisecting step.
3. Repeat step 2 for certain number of times and choose the split for which clustering takes place with maximum overall similarity (Same objective minimization function that is used in the generic k-means algorithm).

4. Repeat Steps 1, 2 and 3 until desired k number of clusters has been reached [15,16].

Steinbach et al. (2000) concluded that the bisecting k-means algorithm outperforms the traditional k-means algorithm in terms of accuracy and efficiency. He also indicated that the k-means algorithm and the bisecting k-means algorithm are better than agglomerative hierarchical clustering. It was determined to be a better approach in terms of the quality of clusters it produces. He highlights one big disadvantage of agglomerative hierarchical clustering. He asserts that mistakes can happen in the earlier stages such as considering the nearest neighbors of a document that belong to a different class and thus cannot be corrected later [17].

Traditional k-means also has several shortcomings:

- The accuracy depends on the estimation of the initial input parameters set for the number of clusters.
- It may be unsuitable to find clusters with different sizes.
- It is sensitive to noise [16]. Noise is defined as unstructured data points or outliers in a data set that are not part of any cluster and decreases the quality of the clustering process.

2.4.3. k-medoid algorithm

To deal with the noise problem, the k-medoid algorithm was developed. It is similar to the k-means algorithm but it does not take the average of all documents in a cluster to find a centroid. The k-mediod algorithm selects one of the documents as its

representative point [18]. It is considered computationally expensive and also does not perform well for large data sets (Kaufman & Rousseeuw, 1990; Krishnapuram, Joshi, & Yi, 1999) [19,20].

2.4.4. Mini-Batch k-means

The computation time of the standard k-means algorithm increases with an increase in the number of documents for large data sets. As such, Sculley D. (2010) proposed a Mini-Batch k-means variant that consumes less computation time for large data sets than the traditional k-means algorithm. It makes use of mini-batches to reduce time, but it uses the same objective function as the k-means algorithm. Mini-batches are small randomly selected samples of input data selected during each iteration, which significantly reduces the computation time.

The main steps in this algorithm are:

1. Select samples from a dataset, which collectively forms a mini-batch.
2. Assign these samples to the nearest centroid, then re-calculate the centroid.
3. Repeat steps 1 and 2 until convergence or an iteration limit is reached [26,14].

Figure 2.6 shows the convergence speed of a mini-batch with k=3 and k=10 as compared to the traditional k-means algorithm and the simple batch k-means algorithm. The results show that the mini-batch converges faster and produces better results even on large data sets [26]. Conversely, the traditional k-means algorithm converges more slowly on large data sets.

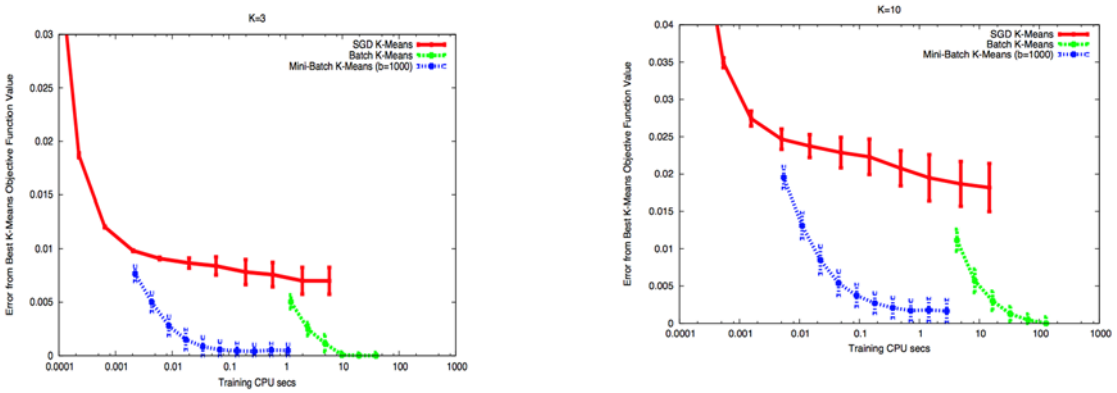


Figure 2.6: Convergence Speed of Mini-Batch with k=3 and k=10 [26]

2.4.5. CLARANS (CLustering Algorithm based on RANdomized Search)

The CLARANS algorithm uses sampling techniques along with PAM (Partitioning Around Medoids). It does not select a fixed sample at a given time, rather it chooses a sample randomly during each and every step of the search. "It treats the clustering process as traversing a graph where every node is a potential solution, that is, a set of k-medoids" [33, 35].

The CLARANS steps for clustering are as follows:

1. Choose random samples of neighbors.
2. Two nodes of the graph are neighbors if they are only one medoid away from each other.
3. Attach a cost for each node to define the total dis-similarity between every object and the medoid of the cluster.
4. Search for a minimum on the graph.
5. If a local optimum is found, then repeat steps 1 through 4.

"The *number of local optimum to search for* is a parameter" [33, 35].

Experimental results show that CLARANS is more efficient than k-medoids and is good at analyzing the outliers [33, 35].

The high-level methodology of CLARANS is shown in figure 2.7 below:

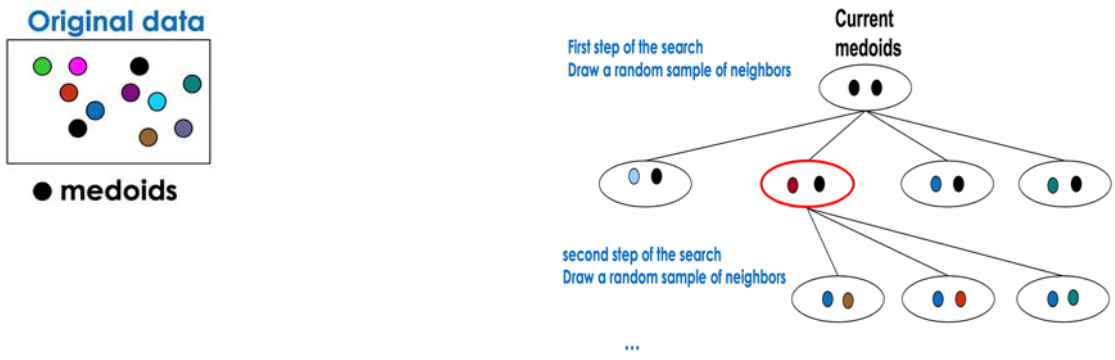


Figure 2.7: CLARANS Algorithm [34]

2.5. Density Based Clustering Methods

The shapes of clusters found by partitioning algorithms are flat geometries. In addition, the accuracy of the output is domain-oriented. Density based clustering methods are efficient and effective in the following ways:

- Less dependency on Domain Knowledge.
- Efficient clusters of arbitrary shape on very large spatial databases.
- Does not require prior information of the number of clusters.
- Handles noise [21].

The general idea is that the clusters are very dense regions separated from lesser density regions.

2.5.1. DBSCAN (Density Based Spatial Clustering of Applications with Noise)

DBSCAN is said to be the most efficient algorithm that works well with large spatial datasets and able to determine noise and outliers from datasets. It creates clusters by finding the densest regions in datasets utilizing two input parameters Eps and MinPts. Eps is defined as the maximum distance between neighborhood points p and q [21]. MinPts defines the minimum number of points in a cluster. The final clusters are of arbitrary shape as shown in figure 2.8 below:

Figure 2.8: Clusters of three different datasets from SEQUOIA 2000 benchmark database [21]

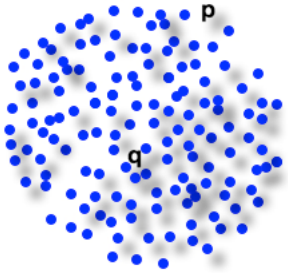
DBSCAN creates clusters in the following way:

a. **The Eps- neighborhood of a point:** A point p is part of a cluster C if there is at least one point q that is closer to it than distance Eps, and $N_{Eps}(q)$ is having at least MinPts points. $N_{Eps}(p)$ is defined as:

$$N_{Eps}(p)= \{q \in C \mid dist\ (p,q) \leq Eps\} \quad (10)$$

b. Density reachable: There are two types of points in a cluster:

- Border Points
- Core Points



p: Border Point

q: Core point

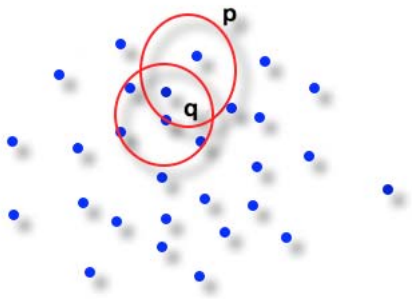
Figure 2.9: Core and Border Points [21]

Figure 2.9 shows that the Eps neighborhood of border points has fewer points than a core point. Therefore, a border point p will be part of cluster if it is closer to any core point q as shown in figure 2.10. The q is considered to be a core point if it has minimum number of points in its neighborhood where:

$$p \in N_{Eps}(q) \quad (11)$$

and,

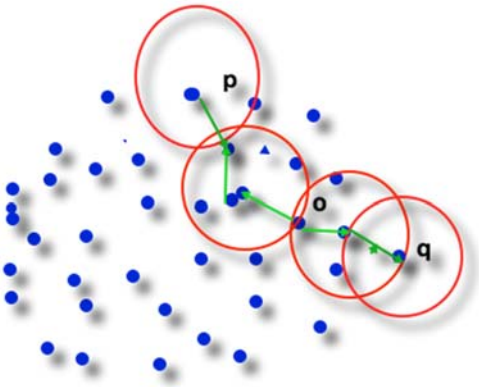
$$|N_{Eps}(q)| \geq MinPts \qquad (12)$$



P is density reachable from q, but not vice versa

Figure 2.10: Density reachable [21]

c. Density Connected: "*A point p is density-connected to a point q with respect to Eps and MinPts if there is a point o such that both, p and q are density-reachable from o with respect to Eps and MinPts*" [21,22].



P and q are density connected via o.

Figure 2.11: Density connectivity [21]

d. **Cluster:** "*If point p is a part of a cluster C and point q is density-reachable from point p with respect to a given distance and a minimum number of points within that distance, then q is also a part of cluster C"* [21,22].

- *"∀p, q: if p ∈ C and q is density-reachable from p with respect to Eps and MinPts, then q ε C"* .

Two points belongs to the same cluster *C*, is the same as saying that *p* is density-connected to *q* with respect to the given distance and the number of points within that given distance.

- *"∀ p, q ∈ C: p is density-connected to q with respect to Eps and MinPts"* [21,22].

e. Noise: All other points that are not part of any cluster are considered noise and not included in a cluster.

2.5.2. OPTICS (Ordering Points to Identify the Clustering Structure)

OPTICS is another density based clustering algorithm for large spatial data sets. It works in a similar way as DBSCAN. However, it overcomes one drawback of DBSCAN in that it discovers clusters in data with varying density. OPTICS orders the points of a data set such that spatially closer points are in the neighborhood. Like DBSCAN, it also requires parameters *Eps* and *MinPts*. In addition, it also takes into

consideration points in more dense clusters so that each point has a Core Distance and a Reachability Distance. "*Core Distance is defined as minimum distance between core object p and other neighborhood object such that this neighbor lies in Eps neighborhood of p . . . On the other hand, reachability distance is the smallest distance from object p from core object o such that p is directly density reachable from o*" [27].

The OPTICS algorithm uses these two parameters to do ordering of data set points and creates density-based clusters. It does not require setting any global parameter but rather uses augmented cluster ordering information. However, this approach has been shown to be slower than DBSCAN [27].

2.5.3. DBCLASD (Distribution Based Clustering algorithm for mining in LArge Spatial Databases)

DBCLASD is another variant of DBSCAN but it does not require any input parameters and that is the biggest advantage of this algorithm over the other approaches. Being non-parametric in nature, its cluster extraction accuracy is efficient and clusters are of good quality and arbitrary shape in large spatial data sets. It is an incremental algorithm and works as indicated below:

1. It does not take into consideration the whole dataset at once; rather it considers points that have been processed so far in order to add points to a cluster.
2. It starts by adding neighboring points to the initial cluster if they are within an expected distance range.

3. Candidates for clusters are found using region queries supported by Spatial Access Methods (SAM). For a point p to be part of cluster C, a circle query having center P and radius m is implemented to find candidate points.

The m should satisfy condition:

$$P(N\ Ndist_C(x) > m) < \frac{1}{N} \qquad (13)$$

or

$$m > \left(\frac{A}{\pi.(1-N^{\frac{-1}{N}})}\right)^{\frac{1}{2}} \qquad (14)$$

where N is the number of points or elements in C and A is the area of C.

This incremental approach implies that:

- The unsuccessful candidate is tested again and not rejected or discarded.
- Points that are part of cluster can be moved to other clusters later.

The testing is performed using:

- Augmentation of current cluster by candidate point.
- The chi-square test is utilized to determine if point is within expected distance range.

The experimental results show that efficiency and quality of DBCLASD is close to DBSCAN but it is slower in comparison [28,29].

2.5.4. DENCLUE (DENsity based CLUstEring) Algorithm

This algorithm is efficient for large multimedia datasets with noise. This approach has a mathematical basis that produces clusters of arbitrary shape for high-dimensional data sets with noise. It requires one input parameter that is the radius (ε) [30]. It has been shown to improve density estimation over DBSCAN and OPTICS [29]. This algorithm involves the following steps:

1. It uses statistical density estimation techniques to estimate the *kernel density*, which provides the *local density maxima value.*
2. Clusters are then formed using this *local density maxima value.*
3. Objects having a small *local density maxima value* are treated as noise.
4. The candidate objects are considered to be part of a cluster using a step-wise hill climbing procedure [29,30].

This algorithm outperforms DBSCAN in terms of efficiency and speed as shown in figure 2.12.

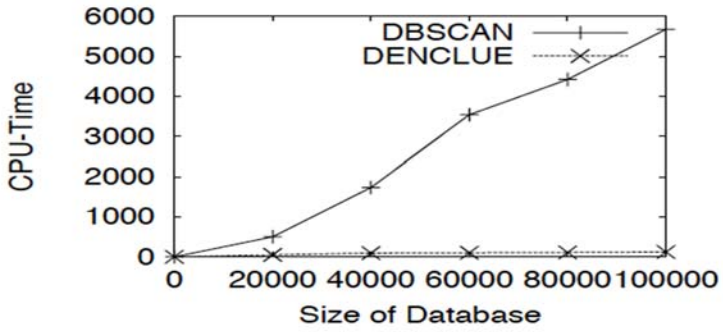


Figure 2.12: Comparison of DENCLUE and DBSCAN [30]

2.6. Term Clustering Methods

These methods are intended to find frequent item sets within data and cluster based upon these items. Some related work in this area is explained below:

2.6.1. Frequent Item Sets Based Clustering

Wang et al. (1999) proposed an approach to cluster text documents by using frequent item sets. This method works by grouping together documents having many frequent items in one cluster and less frequent items in another [23]. Here an item refers to a term or word in the documents.

2.6.2. Hierarchical Frequent Term-based Clustering (HFTC)

HFTC uses a heuristic approach to select the next frequent term to represent the next cluster while minimizing the overlap of clusters. The clustering output strictly depends upon the order of frequent item sets selected (Beil, Ester, & Xu, 2002) [24].

This method produces hierarchical clusters, but does not perform well on large datasets. The reasons are:

- It uses all of the frequent item sets to create the clusters, which could be vary large in size.
- It uses hard clustering such that each document can only be part of at most one cluster (Fung, Wang, & Ester, 2003) [25].

2.6.3. Frequent Item-set based Hierarchical Clustering (FIHC)

To overcome the problems of the HFTC algorithm, FIHC was developed by Fung et al (2003). This algorithm is scalable for large datasets and is "cluster-centered" rather than "document-centered". This means it compares different clusters directly to determine if documents in clusters share more common items than documents in different clusters. This algorithm constructs a tree of documents and labels each cluster accordingly. This algorithm utilizes the concept of a Global Frequent Item (GFI) set. The GFI is defined as *"items that appear more than a minimum fraction of whole document set" [25]*. This helps the algorithm to scale well on large data sets [25].

2.6.4. Term Clustering and Association Rules

This algorithm performs document clustering using AMI (Average Mutual Information) among terms, and Association Rules. The steps in this method are:

1. *Document Participle Process and Pretreatment:* All the documents go through a participle process. All terms are collected and undergo a

pretreatment process. The pretreatment process involves the removal of all far-between and inactive words. The removal of few words having high appearance and replacing words having the same semantic similarity by a formal term.

2. *Term Clustering:* The semantic relevance between two terms T_i, T_j is calculated using AMI as:

$$AMI(T_i, T_j) = \frac{n(T_i, T_j)}{n(T_j)} log \frac{n(T_j)}{n(T_i, T_j)} + \frac{n(T_i, \overline{T}_j)}{n(\overline{T}_j)} log \frac{n(\overline{T}_j)}{n(T_i, \overline{T}_j)} + \frac{n(\overline{T}_i, T_j)}{n(T_j)} log \frac{n(T_j)}{n(\overline{T}_i, T_j)} + \frac{n(\overline{T}_i, \overline{T}_j)}{n(\overline{T}_j)} log \frac{n(\overline{T}_j)}{n(\overline{T}_i, \overline{T}_j)}, 1 \leq i, j \leq m. \quad (15)$$

$n(T_j)$ is the number of documents in which T_j appears

$n(\overline{T}_J)$ is the number of documents in which T_j does not appear

$n(T_i, T_j)$ is number of documents in which T_i, and T_j both appear

$n(\overline{T}_i, \overline{T}_J)$ is number of documents in which neither T_i, nor T_j appear

$n(T_i, \overline{T}_J)$ is number of documents in which T_i, appears not T_j

$n(\overline{T}_i, T_j)$ is number of documents in which T_j, appears not T_i

3. *Term Weight:* The terms are weighted using modified Shannon's reverse document frequency and Dennis's information and noise rate (taken relative frequency of term in consideration instead of absolute frequency).
4. *Vector Space Model:* After term weighing, documents are represented in vector space model as shown:

$$D_k = [(TC_1, r_{k,1}), (TC_2, r_{k,2}), \ldots, (TC_i, r_{k,i}), \ldots, (TC_t, r_{k,t})],\ 1 \leq k \leq n. \quad (16)$$

$r_{k,i}$ is degree of relevance between document D_k and term clustering TC_i.

5. *Use Association Rules to Min Document Clustering:* An association rule is defined as

$$O_p \Longrightarrow O_q \quad (17)$$

where $O_p \subseteq P, O_q \subseteq P, O_p \cap O_q = \Phi$

O is defined as group of objects, and each object is subset of Project set P as written above.

Document similarity is calculated as the square root of the product of the Jaccard Coefficient and the Cosine Coefficient. Moreover, the DHP (Direct Hashing and Pruning)

algorithm is used to minimize item sets and it provides better results than previously explained methods. This method produces comparatively better results in terms of quality and efficiency than k-means and k-mediods. In addition to this, the time cost is less than former techniques [6].

2.7. Core Topic Based Clustering (CTC)

Twitter allows individuals to write short and informal messages called tweets. As such, it is challenging to explore meaningful topics from tweets. Authors have argued that conventional text analysis techniques do not work well on tweet clustering. A Retweet (RT) is a re-posting of someone's tweet. RT helps individuals to share tweets with their followers. In this work, RT is said to contain a representative feature in a tweet.

This method is an extension to term clustering methodology that extracts meaningful topics from tweets using RT and then clusters tweets using these core topics. This method puts more emphasis on RT ratio, which has been used as a weight for the evaluation of clusters. RT is regarded as sign of user's preferences or likes/dislikes. Also it improves cluster quality.

The steps involved in this method are indicated below:

1. Twitter data sets are represented in the form of graph G where each tweet is treated as vertex and each edge describes the similarity relationship between the corresponding vertices.
2. For each seed topic, the corresponding clusters are evaluated as:

$$Eval(C_k) = \frac{\sum_{d_i,d_j \epsilon C_k} sim(d_i,d_j)}{\sum_{d_i \epsilon C_k} \sum_{d_j \notin C_k} sim(d_i,d_j)} \quad (18)$$

where $sim(d_i, d_j)$ is the dot product of the TF-IDF vectors of tweets $d_i, and\ d_j$. TF-IDF is multiplication of TF (Term Frequency) and IDF (Inverse Document Frequency). IDF is the number of tweets in which specific term appears.

3. Pick the top k-clusters based on following condition:

$$arg_{C_k} \max \sum_{C_i \in C_k} w_{C_i} Eval(C_i) \quad (19)$$

where w_{C_i} is ratio of total number of RT in C_i to total number of RT in whole dataset.

The experimental results indicate that it outperforms k-means in terms of speed and accuracy [31].

2.8. Tweets Clustering using hash tags

In twitter, hash tag "#" summarizes the subject of the message being shared. This technique makes use of hash tag co-occurrence frequency to describe the categories of the tweets. This approach includes following steps:

1. Remove tweets with no hash tags to save computational cost.
2. Remove alphanumeric data.
3. Remove overly general hash tags such as *#fb, and # followfriday.*
4. Pick most frequent hash tags and their co-occurrence hash tags.
5. Next stage is to cluster the hash tags using Wu-Palmer distance.
6. Two level filtering techniques are used to minimize noise in hash tags.
7. Calculate similarity measure between hash tag A and hash tag B:

$$S(A,B) = \sqrt{\frac{n_{AB}}{\sum n_{A_j}} + \frac{n_{BA}}{\sum n_{B_j}}} \qquad (20)$$

where n_{ij} is number of co-occurrence between hash tag *i* and hash tag *j*.

8. Use Spectral clustering, Normalized Spectral and METIS to perform tweet-clustering using list of hash tags.
9. Use Cosine Similarity to expand clusters further.

Authors have used different clustering methods to evaluate the performance and the results show that spectral clusters perform better than the other methods [32].

The related work cited in this thesis indicates that much work has been done in this area. This thesis proposes a combinatorial methodology that utilizes agglomerative and divisive hierarchical clustering approaches and the cosine similarity distance measure with feature based clustering. The proposed methodology is compared to k-means, DBSCAN and Ward's Hierarchical Clustering in the following chapters. The results

show improvement over these three methodologies. The proposed methodology in this thesis is explained in greater detail in the following chapter.

3. EXPERIMENATAL SETUP

3.1. Introduction

The review of the literature in the previous chapters indicates that:

- The preprocessing of tweets to reduce dimensionality of features is important for accuracy because it affects the results.
- When hierarchical clustering is applied as a classification tool, there are mistakes that can be made at early stages that may not be corrected at later stages.
- Providing input parameters to clustering algorithms affects the accuracy and type of clusters.
- Similar tweets in each cluster may also be iteratively divided into sub-clusters, which can provide insights of knowledge more efficiently.

The intention of this proposed methodology is to address these items and evaluate the results.

3.2. Proposed methodology

The proposed methodology is shown in figure 3.1 and the steps implemented are explained in following sub-sections.

The proposed methodology consists of four steps:

a. Tweet Pre-processing
b. Feature based Clustering
c. Data Transformation

d. Clustering

i. Intra-cosine similarity based divisive clustering

ii. Inter-cosine similarity based agglomerative clustering

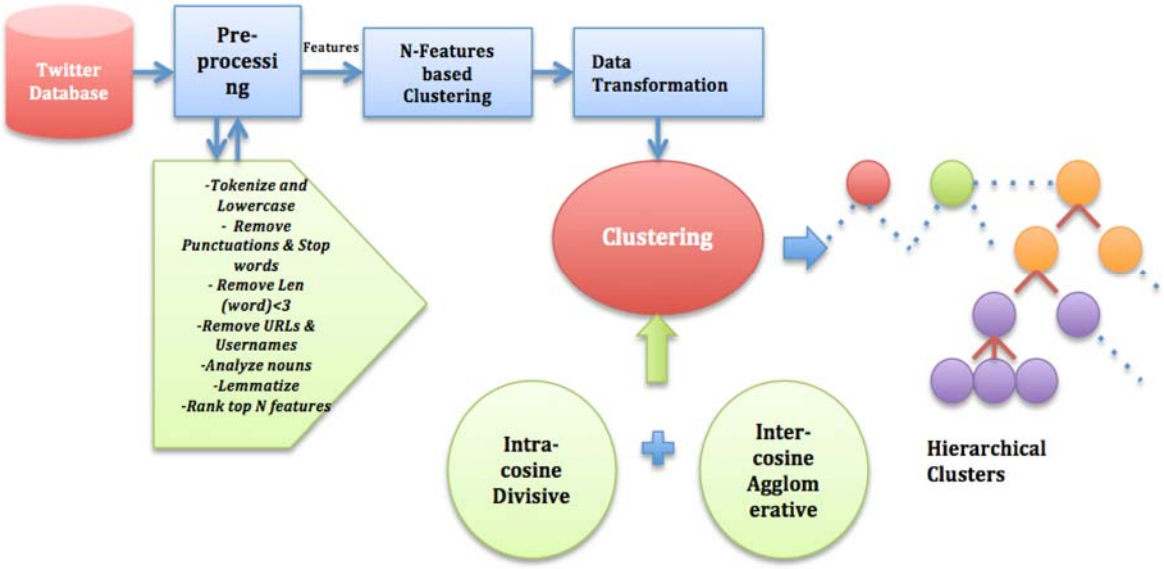


Figure 3.1 Proposed methodology

3.2.1. Tweet Preprocessing

In any given tweet, there can be terms or words that are not significant for consideration in clustering and thus should be removed to retain only the important features. A Tweet is limited to only 140 characters and users tend to use casual language. As such, preprocessing can be a challenging task. However, preprocessing tweets is an important step because it can affect the results of the clustering process.

For example, in text mining or document mining, the most frequent or most important features are utilized. If the dataset is not preprocessed appropriately, the features may not help in mining as shown in the figure 3.2 below.

Figure 3.2 shows the cumulative frequency plot of the 50 most frequent words in *Moby Dick*. There are a total of 19317 samples and 260819 outcomes. The output plot shows that the most frequent words account for approximately half the dataset and none of these words are meaningful. As such, a fine-grained selection of words is required [43].

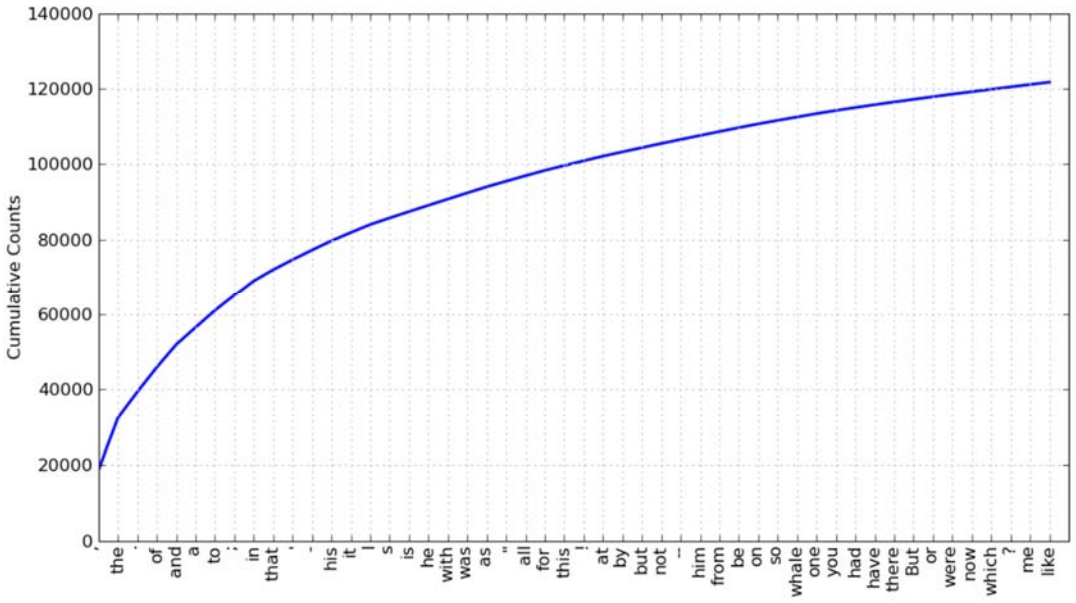


Figure 3.2: Cumulative frequency plot for 50 most frequent words in *Moby Dick [43]*

Various steps have been employed to preprocess tweets to maintain only the meaningful information and are explained in the following sub-sections:

3.2.1.1. *Tokenization*

Tokenization is a process of chopping up a text string or a document into individual pieces knows as tokens.

For example:

"*@AliBunkall : When Obama took office, 180000 US troops were in global conflict zones*"

is tokenized as shown in figure 3.3.

Figure 3.3: Tokenized String

In this methodology, the twitter dataset is tokenized first into small individual pieces or tokens.

3.2.1.2. *Lowercase conversion*

The next step is to change all text into lowercase text so that the same words are treated in the same manner.

For example, *Obama* and *obama* would be treated as two different features if not converted. Therefore, the text is converted to lowercase as shown in figure 3.4 below:

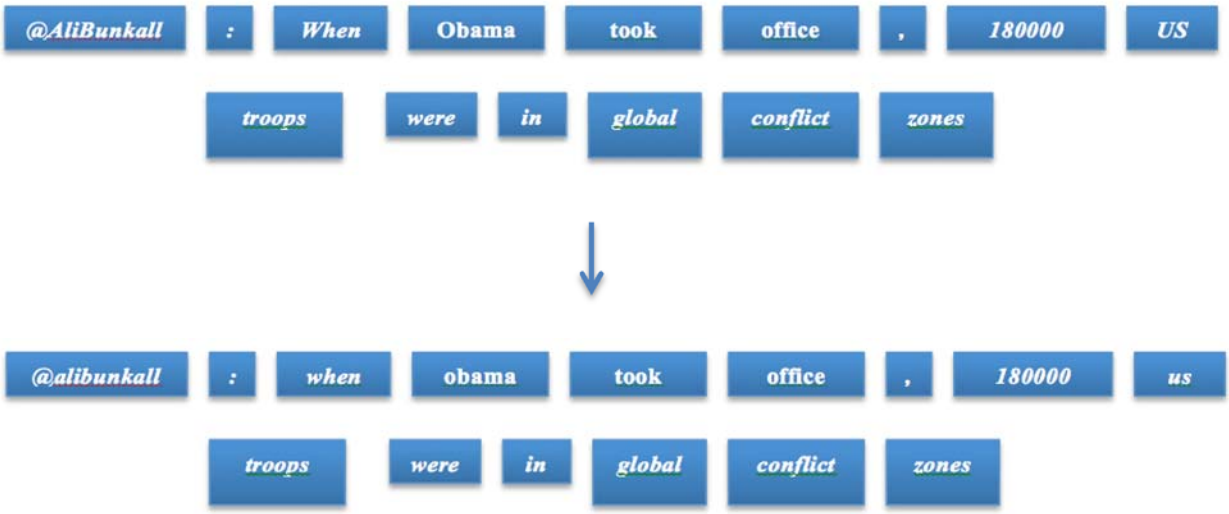


Figure 3.4: Lowercase conversion of tokenized tweet

3.2.1.3. *Punctuation Removal*

As seen in figure 3.2 punctuation makes up a major portion of the word count of any document and is not meaningful for data mining. Therefore, punctuation is also removed.

The figure 3.5 shows the example tweet after punctuation removal.

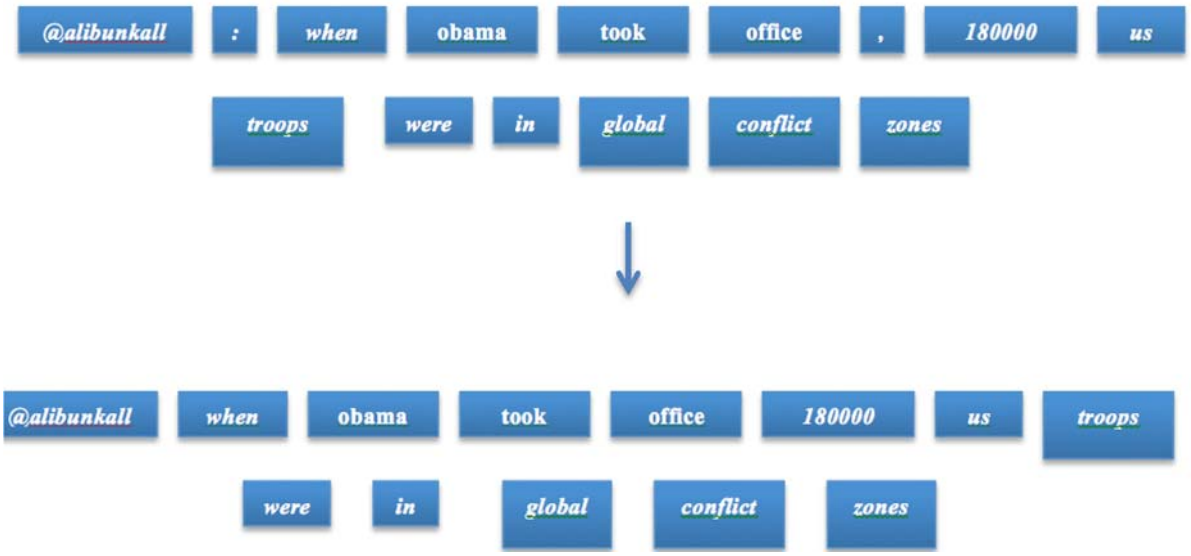


Figure 3.5: Tweet after removing punctuations

3.2.1.4. *Stop words removal*

Stop words are common words that have little lexical content and thus are not significant and should be removed. Therefore, before further processing, stop words are removed to save space and search time. NLTK (Natural Language processing Tool Kit) module can be imported to remove stop words. It consists of 2400 stop words for 11 languages [55]. An example list of stop words in English is shown below:

“ 'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now'” [56].

The example tweet is shown in figure 3.6 below after stop word removal.

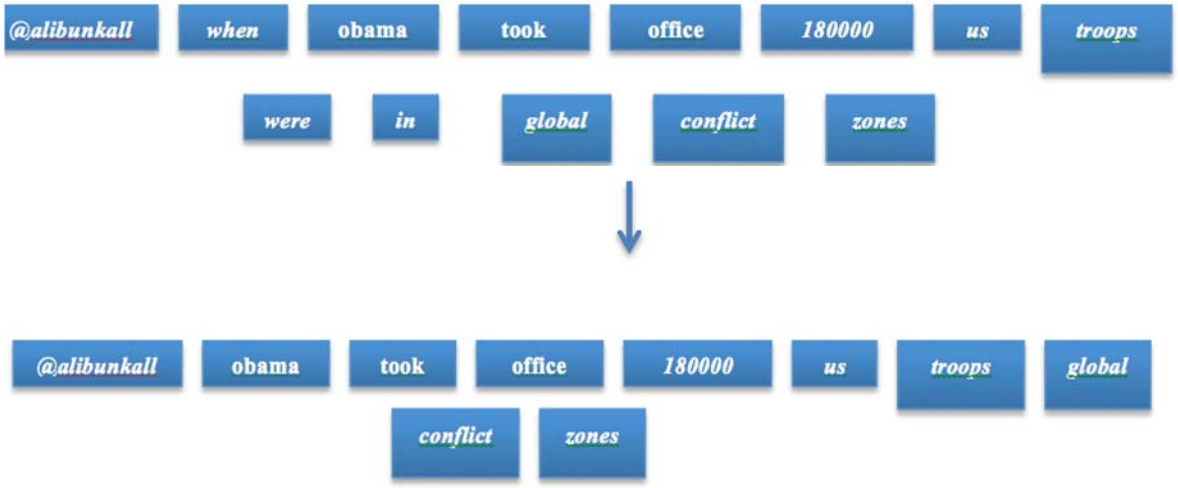


Figure 3.6: Tweet after removing stop words

3.2.1.5. *Short length and alphanumeric words removal*

After going through the previous steps, there may still be some un-important words. Specifically, words with lengths less than three characters. In some cases, short words can be relevant as well. However, in this scenario, the short length features are not

relevant for clustering process and should be removed. As such, further filtration is required to remove words with length less than three characters.

Sometimes alphanumeric words such as “abc123”, “180000” also occur in documents or tweets, that are not significant for clustering. These words are also filtered out. The figure 3.7 shows the tweet after short length word and alphanumeric word removal.

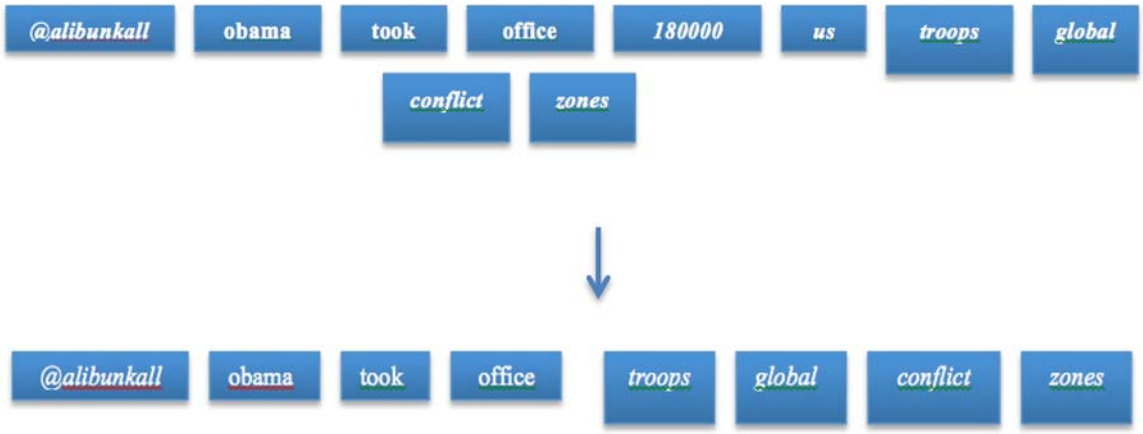


Figure 3.7: Tweet after removing Short length and alphanumeric words

3.2.1.6. *Remove URLs and Usernames*

For Twitter, URLs and usernames are common and do not provide value to clustering so they are also removed. They are not meaningful as unique features or to describe semantics. Usernames start with “@” symbol and URLs with “http”.

Example:

Figure 3.8: Tweet after removing URLs and Usernames

After performing these six steps, tweets will be clean from most of the irrelevant information and can now be used for the feature extraction process.

3.3. Feature Extraction

Features are defined as the unique words or terms in the data set that facilitate the clustering process effectively and efficiently. Feature extraction is important for improved data analysis, model interpretability, reduced training time, reduced searching process and enhanced generalization. Prior to feature extraction, a few more steps are performed as explained in sub-sections below:

3.3.1. *Lemmatization*

Lemmatization is the process of converting different inflected forms of a word into its base form so that they are processed as the same object. For instance, a word "walk" may occur in several inflected forms such as "walked", "walks", "walking", but its base form is "walk". Therefore, lemmatization has been applied on dataset containing cleaned tweets from step (3.1) to remove inflected forms of same word.

An example of lemmatization is shown in figure 3.9 below:

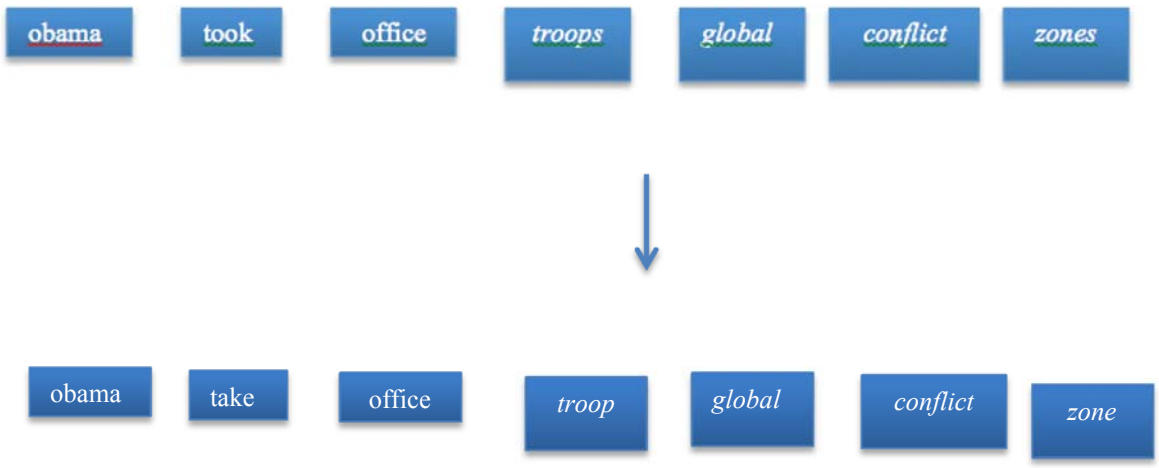


Figure 3.9: Lemmatized text

3.3.2. Part of speech (POS) tagging

The process of tagging words for their corresponding parts of speech is called as POS tagging or lexical categories. There are different kinds of POS tags in the English language and some of these are shown in figure 3.10 below with their description.

In this system, only nouns will be considered as features because nouns are the most meaningful entities among all other words (verbs, adverbs and adjectives are only used to define relationships between noun phrases). Therefore, all other terms are discarded and only nouns are retained (different forms of nouns are shown as highlighted text in figure 3.10).

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word

6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Pre-determiner
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	UH	Interjection
26.	VB	Verb, base form
27.	VBD	Verb, past tense
28.	VBG	Verb, gerund or present participle
29.	VBN	Verb, past participle
30.	VBP	Verb, non-3rd person singular present
31.	VBZ	Verb, 3rd person singular present

Figure 3.10: Penn tree bank POS tags [44]

For example, the figure 3.11 displays tokenized, cleaned and lemmatized tweet, which is to be POS tagged to select proper nouns only.

Figure 3.11: POS tagged tweet

From this step forward, only nouns are retained and these are shown is figure 3.12.

Figure 3.12: Noun features

There are only 5 nouns in the string of 15 words, which accounts for one third of the total words.

3.3.3. *Ranking*

The next step in the process is to count the frequency of all nouns in the dataset. This is also referred as term frequency. This methodology rank the top n features $f_1, f_2, \ldots, f_n$ on the basis of frequency of occurrence. This step significantly reduces the size of feature set and facilitates improved clustering. This is the final step before the clustering process.

3.4. Clustering

Clustering is an automated or semi-automated un-supervised machine learning technique that enables the grouping of similar text documents into similar categories. Text clustering helps make information retrieval and the text browsing process easier and more effective. The various steps under this process are:

3.4.1. *Feature based Clustering*

The clustering process starts by creating broad categories of tweets based on the top n ranked features from the feature set. Thus, there are total of n clusters initially ($C_1, C_2, \ldots, C_n$) for the top n ranked features based on the following definition:

Definition: If $f_n \in t_i$, then $t_i \in C_n$, where C_n is the nth cluster and t_i can be any tweet from dataset that is being compared against f_n. Then there are n clusters $C = \{C_1, C_2, \ldots, C_n\}$.

According to this definition, each feature is compared with tweets in dataset and based on similarity different tweets are collected under one category to form a cluster. This is the first step in clustering process.

3.4.2. *Data transformation*

The next step in the procedure is to represent tweets in an m-dimensional vector form so that the similarity between different tweets can be calculated. The tweets are represented in an m-dimensional vector in the form of term-document matrix. A document vector represents a document as a bag of words.

Let M be the term-document matrix as shown in figure 3.13. Suppose dataset contains n documents in total and have m features in total then the matrix will have n rows (for each document) and m columns (for each unique feature). The entries in the matrix are the frequencies of features in the documents. For instance, entry w_{ij} in matrix M is the frequency of i^{th} feature f_i in j^{th} document d_j [45].

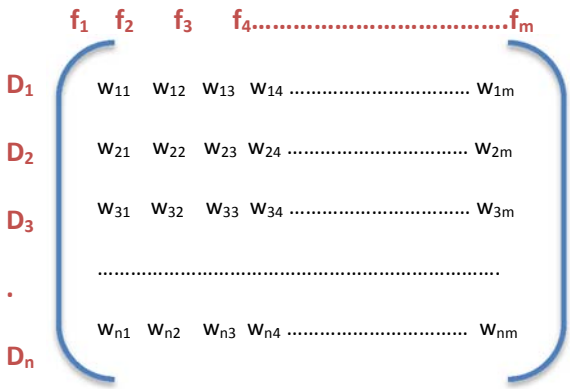


Figure 3.13: Vector Space Model (Term-document matrix) [45]

Tweets in Vector Space Model (VSM) are represented as an m-dimensional vector:

$$\vec{f_t} = (tf(t, f_1), tf(t, f_2), \dots, tf(t, f_m)) \quad (21)$$

where $tf(t, f)$, is term frequency of feature $f \epsilon F$ in tweet $t \epsilon T$. F is the set of features and T is set of all the tweets in the data set.

In practice, some frequent features are not important. To trim those features, a TF-IDF (Term Frequency- Inverse Document Frequency) weighing technique is used [46].

TF-IDF is multiplication of TF and IDF. IDF of a feature f in tweet T is calculated using formula:

$$\text{IDF(f,t)} = log \frac{N}{|\{t \epsilon T : f \epsilon t\}|} \quad (22)$$

where N is the total number of tweets, and $|\{t \epsilon T : f \epsilon t\}|$ is number of tweets which contains feature f. So TF-IDF score of features is calculated as TF*IDF [57].

Thus, a tweet is represented as an m-dimensional vector of TF-IDF weights:

$$\vec{f_t} = (tfidf(t, f_1), tfidf(t, f_2), \ldots, tfidf(t, f_m)) \qquad (23)$$

Now, the VSM of twitter dataset is:

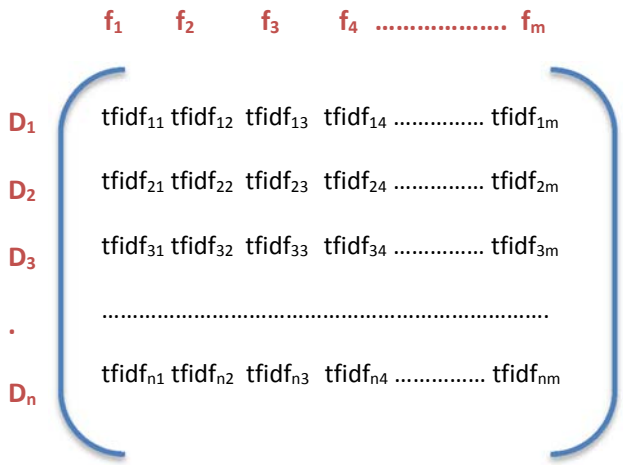


Figure 3.14: New Vector Space Model (Term-document matrix)

3.4.3. *Intra-cosine similarity based Divisive Clustering*

After transforming all the broader categories of clusters in term-document matrix form (VSM), the next step is to compare the tweets under each category for similarity and dividing clusters accordingly so that the most similar tweets are in one cluster. This approach is divisive because clusters will be divided down the hierarchy in order to make the most similar tweets part of different clusters. The various steps in this process are explained below:

a. Calculate centroid: Calculate the mean of all tweets ($\forall t \in C$) belonging to cluster C to figure out the centroid using formula:

$$C_{mean} = \frac{\sum_{t \in C} t}{|C|} \qquad (23)$$

where t is any tweet that belongs to cluster C and $|C|$. The resultant is the mean of all the tweets and the centroid of each cluster.

b. ***Calculate Intra-cosine similarity:*** Each tweet is compared to *Cmean* through an angle θ determined by [36,37]:

$$\theta = \cos^{-1}\left(\frac{\vec{t} \cdot \overrightarrow{C_{mean}}}{|\vec{t}| * |\overrightarrow{C_{mean}}|}\right) \qquad (24)$$

This results in a list of angles for each cluster. Tweets having angle greater than the average angle in the cluster is considered dis-similar and becomes part of new cluster. This continues iteratively to form a tree until there are no more dis-similar clusters.

c. Boundary Condition: The iterative algorithm continues until a given child node is a null set (as shown in figure below). If a child node is a null set, then repeat step a, b and c until all the cluster categories are finished.

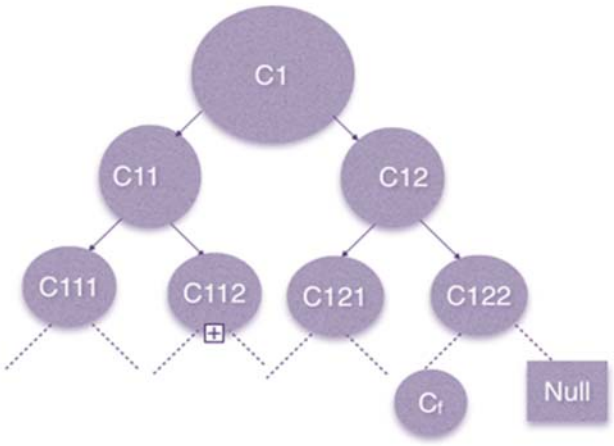


Figure 3.15: Boundary Condition

The leaf nodes of this tree are the final clusters.

3.4.4. *Inter-cosine similarity based Agglomerative clustering*

It is possible that clusters may contain similar information or duplicate information. To remove this, the similarity of any two clusters from the previous steps is calculated in order to find out their similarity score.

This is a bottom up procedure that works as explained below:

a. ***Choose Clusters:*** The very first step is to choose two clusters for similarity comparison.

b. ***Calculate Inter-cosine similarity:*** Calculate the angle θ for two clusters chosen in step (a) as [36,37]:

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{C_{mean1}} \cdot \overrightarrow{C_{mean2}}}{\left|\overrightarrow{C_{mean1}}\right| * \left|\overrightarrow{C_{mean2}}\right|}\right) \qquad (25)$$

where *Cmean1* is Centroid of cluster1 and *Cmean2* is centroid of cluster2, that

will return an angle. This step provides the angular similarity of clusters with each other.

c. ***Merge or remove redundancy:*** The angle calculated in previous step is compared and is determined to fall under one of the following categories. A decision is made according to the conditions in table 1.

Angle	Decision
$0 \leq \theta \leq 15$	Tweets are extremely similar – Merge and remove Redundancy
$15 < \theta \leq 30$	Tweets are very similar- Merge and remove Redundancy
$30 < \theta \leq 45$,	Tweets are mostly similar- Merge and remove Redundancy
$45 < \theta \leq 60$	Tweets may be similar- Don't merge
$60 < \theta \leq 90$,	Tweets are dis-similar- Don't merge

Table 3.1: Decision table for merge or remove operation

If any inter-cosine similarity score falls under first three categories, those clusters will be merged and redundant tweets will be deleted as shown in figure 29. This step produces the final clusters.

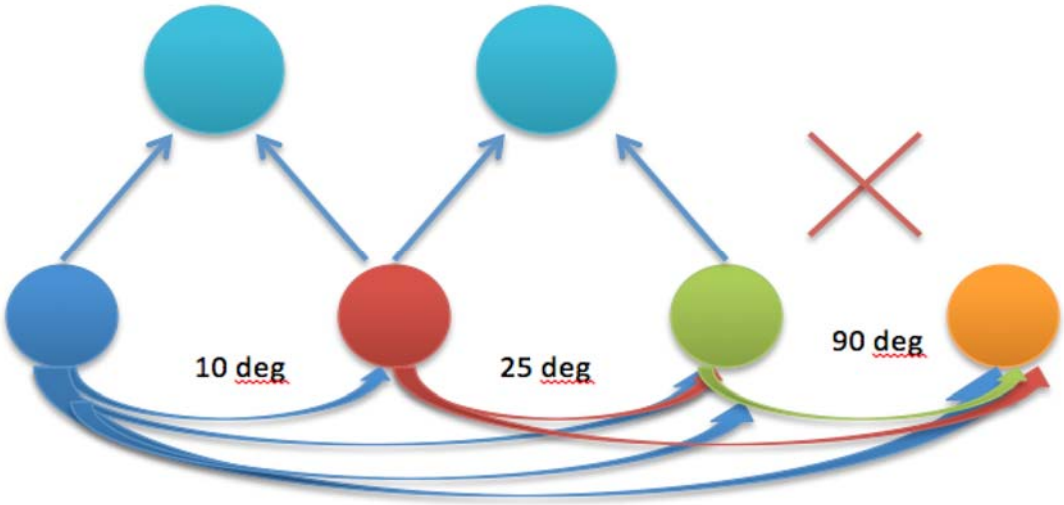


Figure 3.16: Merging and removing redundancy

d. Boundary Condition: Repeat steps a through c until all the clusters are compared against each other. After following this procedure, final clusters are formed.

This approach has been tested and compared against existing methodologies and is shown to produce promising results. It works well with large data sets. However, it may be slower than other approaches because of the combination of two hierarchical clustering approaches.

4. EXPERIMENTAL RESULTS

This chapter shows the results of proposed methodology on real time tweets collected from Twitter API. It automatically divides the tweets into meaningful clusters. The results are then compared to existing methodologies.

4.1. Stem Cell Tweets

A Twitter dataset was collected from the Twitter API to be used for the experimental clustering of tweets. Tweets were gathered from Twitter using a "Stem Cell" query to the Twitter API. The tweets were cleaned of SPAM and subsequently used for semantic analysis. The dataset contains 15062 tweets.

Dataset	No. of tweets
Stem Cell	15062

Table 4.1: Dataset used

4.2. Preprocessing tweets

After the dataset was gathered, the tweets were preprocessed to remove punctuation, stop-words, hash-tags, user names and URLs as described in Chapter 3. The nouns were extracted from the twitter dataset and used as features for analysis. There were determined to be a total of 8791 features in the Stem Cell dataset.

Dataset	No. of tweets	No. of features
Stem Cell	15062	8791

Table 4.2: Number of features in dataset

4.3. Ranking and Feature based Clustering

4.3.1. Rank features

After getting all the features, they were ranked according to their corresponding frequency and top k nouns were selected according to the following equation:

$$k = \frac{Number\ of\ features}{3(Number\ of\ tweets)}\ \% \qquad (26)$$

k determines the number of clusters the system will initially start clustering with. Table 4.3 shows that this equation provides user with reasonable value of k to start with.

Dataset	No. of Tweets	No. of features	k
Pepsi	38576	19474	17
Coke	50220	20791	14
Katy Perry	11359	5267	15
Obama	10440	6052	19
Nokia	18708	11659	21

Table 4.3: Experimental stats for k

This thesis uses Stem Cell tweets gathered over a period of three months. Table 4.4 shows the results of tweets gathered and the value of k.

Dataset	No. of tweets	No. of features	k
Stem Cell	15062	8791	19

Table 4.4: Top k features

The k features analyzed in this experiment are shown in figure 4.1 below:

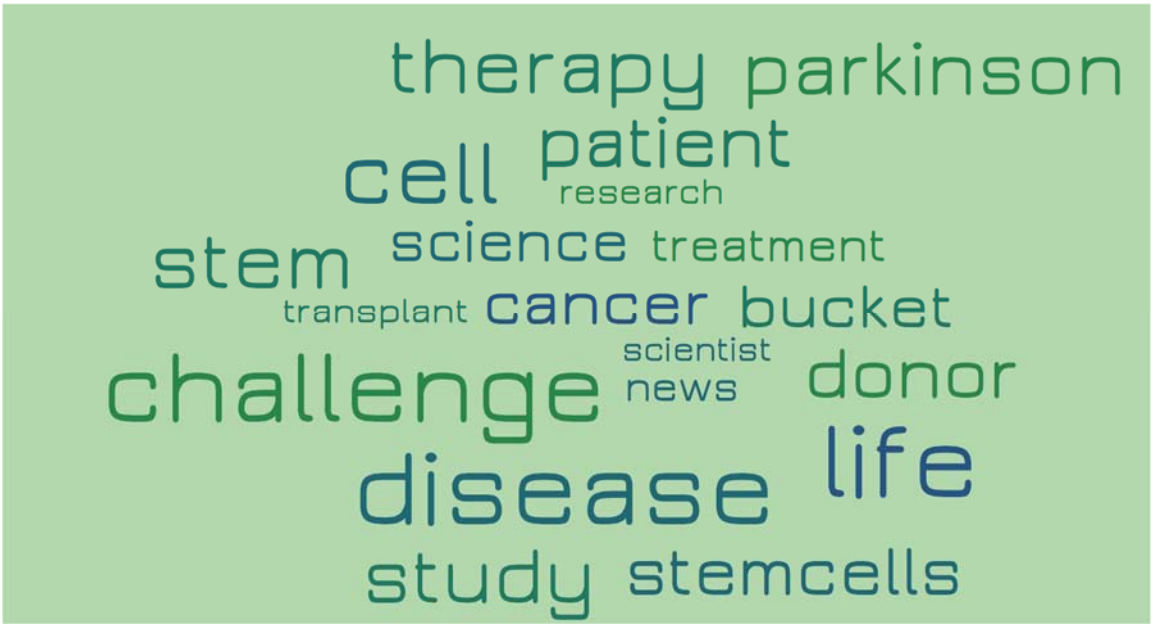


Figure 4.1: Top k features

These k features are shortlisted based upon the frequency of their occurrence. Table 4.5 shows features and their frequency of occurrence in given twitter data set.

Features	Frequency of Occurrence
Cell	13810
Stem	13619

Research	3375
Treatment	1631
Therapy	1368
Transplant	1293
Parkinson	713
News	638
Donor	627
Challenge	555
Scientist	444
Study	507
Bucket	503
Patient	481
Life	468
Stemcells	444
Disease	432
Cancer	420
Science	400

Table 4.5: Top k features and their frequency of occurrence

4.3.2. Feature based clustering

After determining the top k features, tweets were clustered under these categories. This was the first step to create broader categories based on the most frequent nouns.

Figure 4.2 shows some sample tweets from *Treatment* cluster:

Treatment
new stem cell operation could revolutionise treatment of knee injuries http://t.co/ri922lezlh via @guardian
hair regrowth treatment stem cell hair restoration technique: hair regrowth treatment stem cell hair restorati... http://t.co/tlb8zzxyku
rt @philippinebeat: i disagree with the good doh sec. stem cell treatment is not good for all diseases. and why do we want to be in the for
rt @philippinebeat: we have to be careful w/ stem cell treatments. for ex if you culture stem cells for a cancer patient & you are not care
new stem cell operation could revolutionise treatment of knee injuries http://t.co/aee1qcmvyl
hair regrowth treatment stem cell hair restoration technique http://t.co/7jb2kb1inc
@team_inquirer @leisalaverria the lady is right. stem cell treatment is only for the rich not the poor filipinos.
promising #southampton #abicus #stemcell #knee injury treatment regenerates damaged tissue http://t.co/vks5fz3ucp #sportsinjury
rt @vsinghhh: meditation rather than medication: discussed stem cell research & new treatments. consensus: we prefer prevention! http://t.c
news: #stemcells a new stem cell operation could revolutionise treatment of knee injuries http://t.co/qpiqj4viux

Figure 4.2: Sample tweets from treatment Cluster [Feature based clustering]

4.4. Divisive intra-cosine similarity clustering

After obtaining the broader categories, the next step is to apply divisive intra-cosine similarity clustering on each cluster. This results in dividing each cluster into two sub-clusters. One contains the most similar tweets and the other contains the dissimilar tweets from the former cluster.

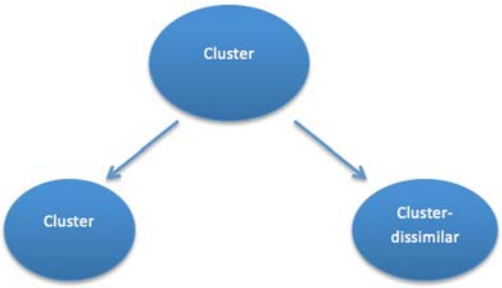


Figure 4.3: Divisive intra-cosine similarity clustering

After divisive intra-cosine similarity clustering, the total clusters become 38.

Dataset	No. of tweets	No. of features	k	Divisive clusters
Stem Cell	15062	8791	19	38

Table 4.6: Number of divisive Clusters

As such, there will be two sub-clusters for each broad category. The second sub-category contains the dissimilar tweets from the original file based on cosine similarity score.

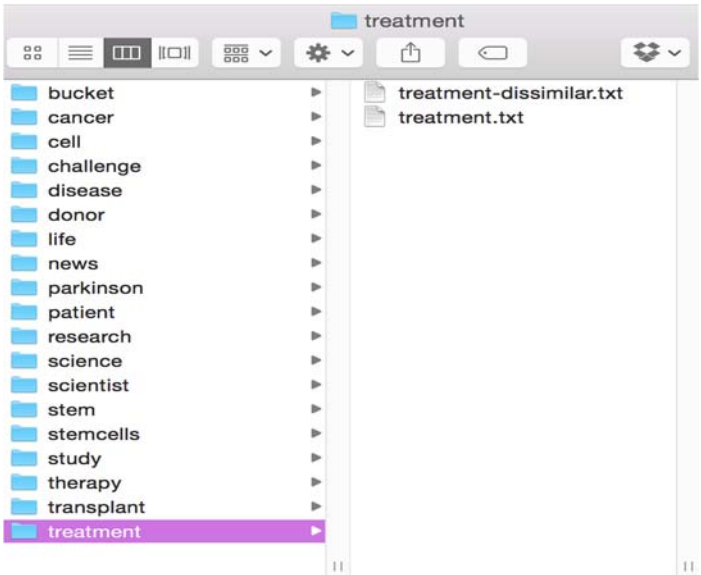


Figure 4.4: Divisive Clustering Output

Some of the sample tweets of treatment category are shown in figures 4.5 and 4.6 below:

Treatment
hair regrowth treatment stem cell hair restoration technique: hair regrowth treatment stem cell hair restorati... http://t.co/tlb8zzxyku
rt @philippinebeat: i disagree with the good doh sec. stem cell treatment is not good for all diseases. and why do we want to be in the for
rt @philippinebeat: we have to be careful w/ stem cell treatments. for ex if you culture stem cells for a cancer patient & you are not care
hair regrowth treatment stem cell hair restoration technique http://t.co/7jb2kb1inc
@team inquirer @leisalaverria the lady is right. stem cell treatment is only for the rich not the poor filipinos.
promising #southampton #abicus #stemcell #knee injury treatment regenerates damaged tissue http://t.co/vks5fz3ucp #sportsinjury
rt @vsinghhh: meditation rather than medication: discussed stem cell research & new treatments. consensus: we prefer prevention! http://t.c
rt @thewayofjay: #geekspeak 8yrs after a stem cell treatment to treat paralysis, a woman started growing a working nose on her spine. http:
i would love to try stem cell treatment for my lower back pains. i hear it's pretty established in germany. anyone know anything about this?
hair regrowth treatment stem cell hair restoration technique http://t.co/oc0k4nsmbr
new biomarkers found may lead to developing stem cell replacement treatments in the inner ear. #audiology #audpeeps http://t.co/wgdamykfgv
bioheart announces world's first combination stem cell treatment http://t.co/0kpwn1bc1e #biotech

Figure 4.5: Sample tweets from treatment category [Divisive clustering]

Treatment-dissimilar
new stem cell operation could revolutionise treatment of knee injuries http://t.co/ri922lezlh via @guardian
new stem cell operation could revolutionise treatment of knee injuries http://t.co/aee1qcmvyl
news: #stemcells a new stem cell operation could revolutionise treatment of knee injuries http://t.co/qpiqj4viux
news: #stemcells bioheart announces world's first combination stem cell treatment

http://t.co/mshalh6gou

ms stem cell therapy treatment hope for mum http://t.co/0ugfatridg

stem cell transplants show promise for treatment of parkinson's disease: http://t.co/y3ozxydnxm via @mocost #parkinsons #medicine #neuro

rt @simba37: new stem cell operation could revolutionise treatment of knee injuries http://t.co/5tjoe0ror9 via @guardian --a way to finally

amazing how non-fda approved stem cell treatments helped this woman walk again. http://t.co/59m5pg8jiw #inspirational #stemcell #healthcare

new stem cell operation could revolutionise treatment of knee injuries http://t.co/uds8dboxck via @guardian

Figure 4.6: Sample tweets from treatment-dissimilar category [Divisive clustering]

This continues iteratively to form a tree until there are no more dis-similar clusters. Figure 4.7 shows the iterative algorithm that continues until a given child node is a null set.

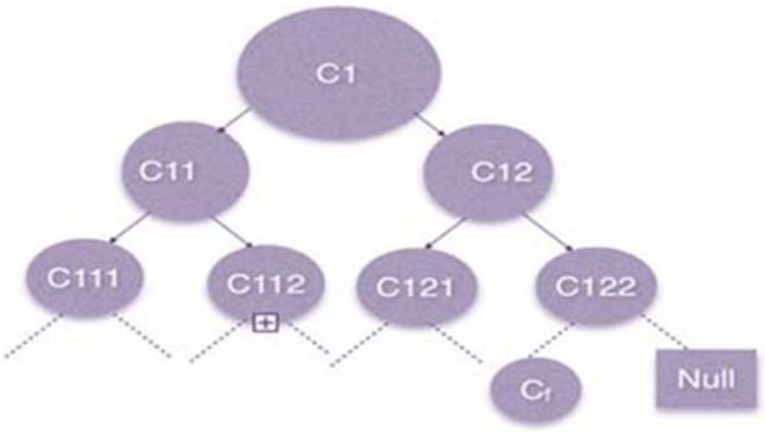


Figure 4.7: Iterative algorithm Boundary Condition

This iterative process gives sub-clusters containing the most similar tweets. However, some of the clusters may contain redundant tweets. Once divisive clustering was completed there were 101 total clusters that also included redundant clusters.

Dataset	No. of tweets	No. of features	K	Total Divisive clusters
Stem Cell	15000	8791	19	101

Table 4.7: Total number of divisive clusters

4.5. Agglomerative inter-cosine similarity clustering

The last step in this proposed methodology is to traverse clusters named tree upwards so as to calculate inter-cosine similarity of divisive clusters with each other to merge the most similar ones together to increase effectiveness of clustering process and remove redundancy. After applying this approach, total clusters in the end would be 60.

Dataset	No. of tweets	No. of features	K	Divisive clusters	Agglomerative Clusters
Stem Cell	15062	8791	19	101	60

Table 4.8: Total number of agglomerative clusters

The number of final categories after this step is 10 (shown in table 4.8 below):

Dataset	No. of tweets	No. of features	No. of categories	No. of Clusters
Stem Cell	15062	8791	10	60

Table 4.9: Total number of final categories and clusters

The final categories and sub-clusters are shown in figures 4.8 and 4.9 below:

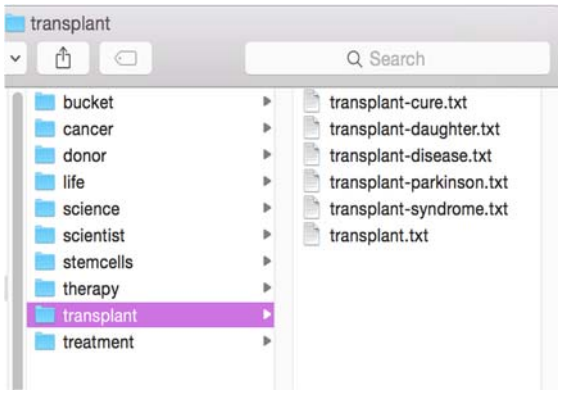


Figure 4.8: Iterative process of creating sub-clusters [transplant]

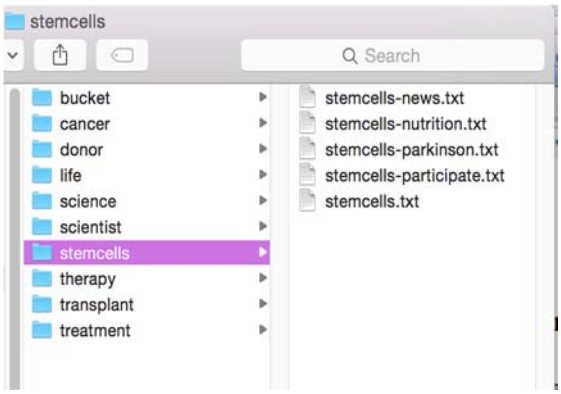


Figure 4.9: Iterative process of creating sub-clusters [stemcells]

Some of the Tweets in Treatment category are shown in figures 4.10, 4.11 and 4.12:

Treatment-knee injuries
new stem cell operation could revolutionise treatment of knee injuries http://t.co/ri922lezlh via @guardian
new stem cell operation could revolutionise treatment of knee injuries http://t.co/aee1qcmvyl
news: #stemcells a new stem cell operation could revolutionise treatment of knee injuries http://t.co/qpiqj4viux
news: #stemcells bioheart announces world's first combination stem cell treatment http://t.co/mshalh6gou
ms stem cell therapy treatment hope for mum http://t.co/0ugfatridg
stem cell transplants show promise for treatment of parkinson's disease: http://t.co/y3ozxydnxm via @mocost #parkinsons #medicine #neuro
rt @simba37: new stem cell operation could revolutionise treatment of knee injuries http://t.co/5tjoe0ror9 via @guardian --a way to finally

Figure 4.10: Treatment-knee category sample tweets

Treatment-back pain
nadal to receive stem cell treatment on back (via http://t.co/ia0ypecygb)

http://t.co/qrzz5nizqe
rt @tennischannel: nadal to receive stem cell treatment for back pain http://t.co/fifrsuee9u
rafael nadal will undergo stem cell treatment on his back http://t.co/oygu2h42mt via @sinow
nadal to receive stem cell treatment for back pain http://t.co/n7xpfx54ly
nadal to get stem cell treatment on back http://t.co/puxwipkbdc via @yahoosports
@tennischannel: nadal to receive stem cell treatment for back pain http://t.co/zpmzfwmqrz @phenomanun
@espntennis: rafael nadal to receive stem cell treatment for back pain http://t.co/fuc4jn2qhy

Figure 4.11: Treatment-back pain category sample tweets

Treatment stem cell
hair regrowth treatment stem cell hair restoration technique: hair regrowth treatment stem cell hair restorati... http://t.co/tlb8zzxyku
rt @philippinebeat: i disagree with the good doh sec. stem cell treatment is not good for all diseases. and why do we want to be in the for
rt @philippinebeat: we have to be careful w/ stem cell treatments. for ex if you culture stem cells for a cancer patient & you are not care
hair regrowth treatment stem cell hair restoration technique http://t.co/7jb2kb1inc
@team_inquirer @leisalaverria the lady is right. stem cell treatment is only for the rich not the poor filipinos.
promising #southampton #abicus #stemcell #knee injury treatment regenerates damaged tissue http://t.co/vks5fz3ucp #sportsinjury
hair regrowth treatment stem cell hair restoration technique http://t.co/ttplgjgthb

Figure 4.12: Treatment-stem cell category sample tweets

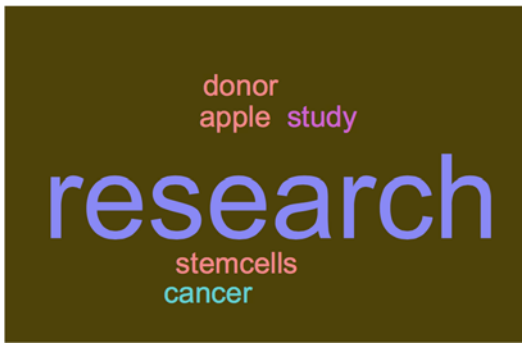
4.6. Comparison with existing techniques

The proposed methodology's results are compared with three popular document-clustering techniques i.e. k-means, Ward's Hierarchical Clustering and DBSCAN.

4.6.1. k-means

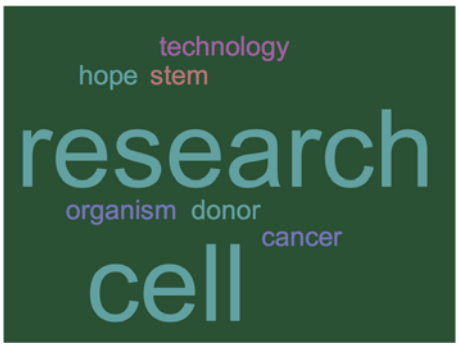
As explained earlier, k-means requires user to input number of clusters i.e. k in

advance. So, the experiment is performed with k=7 and k=10, and the categories that emerged are shown in figures 4.14 and 4.15 with word clouds that represent the number of tweets in each category respectively.



k=7

Figure 4.13: Categories with k=7 [k-means]



k=10

Figure 4.14: Categories with k=10[k-means]

In both of the experiments, it has been observed that for the Stem Cell Data Set, it clustered a very large number of tweets (approximately 85% of all the tweets) under the "Research" category, which is arguably not very useful if the vast majority of tweets are identified to be in the same category. Moreover, most of the tweets do not seem to be relevant to this category. For example, the highlighted tweets in figure 4.16 are related to "treatment" category, but these tweets appear under wrong category. It can be faster than proposed methodology but the clusters are not of good quality.

Research
New stem cell operation could revolutionise treatment of knee injuries http://t.co/Ri922lEZLh via @guardian
RT @SaveMaiDuong: At 7:10! RT @CTVCanadaAM: Tmr we speak with #MaiDuong ... the MTL woman in desperate need of Vietnamese stem cell donors.

I'm raising money for Stem Cell Therapy for Zedd.... Click to Donate: http://t.co/rl0d0dttpg #gofundme

Hair Regrowth Treatment Stem Cell Hair Restoration Technique: Hair Regrowth Treatment Stem Cell Hair Restorati... http://t.co/tLb8ZzxYKu

RT @Shawndoyle: PLEASE RT! She has 6 weeks to find donor. Montreal woman desperately seeks Vietnamese stem cell donors http://t.co/Cu98zrN

Laurence Chilcott Press Releases : Adult Stem Cell Advocate and Stem Cell Nutrition Expert .. - See on... http://t.co/Gf67I3TgiX

@Team_Inquirer @leisalaverria THE LADY IS RIGHT. STEM CELL TREATMENT IS ONLY FOR THE RICH NOT THE POOR FILIPINOS.

RT @Shawndoyle: PLEASE RT! She has 6 weeks to find donor. Montreal woman desperately seeks Vietnamese stem cell donors http://t.co/Cu98zrN

Stem cells: Taking a stand against pseudoscience http://t.co/0yCoDRlC78
Serum Stem Cell... http://t.co/VQ67gf1fys

RT @PhilippineStar: Sen. Nancy Binay asks Abad about #DAP funds used in stem-cell research. DOH Sec. Ona told to explain. | via @xtinamen

RT @thewayofjay: #GeekSpeak 8yrs after a stem cell treatment to treat paralysis, a woman started growing a working NOSE on her spine. http:

RT @xtinamen: Sen. Nancy moves from stem cell to COA projects funded by DAP.

RT @ABSCBNNews: DOH Sec. Ona explains priority given to stem cell research project (given P70M). Says will provide committee with findings

RT @Team_Inquirer: Sen. Binay questions stem cell project funded by DAP, saya buying hospital beds more urgent. | @leisalaverria

RT @PawieSharpei: With all due respect, Sec. Ona, stem cell research is still in its INFANCY and hence not a priority for a resource-scarce

Sorry but I think the stem cell research was really important.

New stem cell operation could revolutionise treatment of knee injuries http://t.co/aEE1qcMVYL

Figure 4.15: Sample tweets of research category [k-means]

4.6.2. Ward's Hierarchical Clustering

Similarly, for the Ward's Hierarchical Clustering methodology using the stem cell data set with 10 clusters, the categories emerged as shown in figure 4.17 and the vast

majority of tweets were clustered under one category, “Research”. Moreover, there were 4 clusters for “treatment” category only.

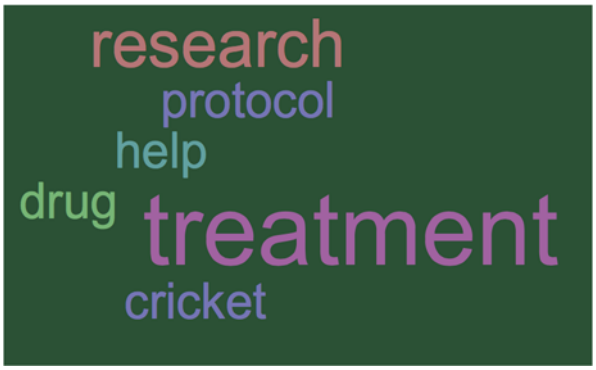


Figure 4.16: Categories when number of clusters are 10 [Ward’s algorithm]

The quality of the clusters is also arguably not good. Most of the tweets clustered under the “research” category are irrelevant. The highlighted tweets in figure 4.18 should appear in treatment category.

Research
new stem cell operation could revolutionise treatment of knee injuries http://t.co/ri922lezlh via @guardian
rt @savemaiduong: at 7:10! rt @ctvcanadaam: tmr we speak with #maiduong ... the mtl woman in desperate need of vietnamese stem cell donors.
i'm raising money for stem cell therapy for zedd.... click to donate: http://t.co/rl0d0dttpg #gofundme
rt @cdnstemcell: an editorial in nature offers hope for stem cell science and explains why it takes so long get cures to clinics: http://t.
younger looking healthy skin all begins with your at home care, the max stem cell cleanser is youth in a bottle.... http://t.co/ufgn7ahugx
hair regrowth treatment stem cell hair restoration technique: hair regrowth treatment stem cell hair restorati... http://t.co/tlb8zzxyku
rt @shawndoyle: please rt! she has 6 weeks to find donor. montreal woman desperately seeks vietnamese stem cell donors http://t.co/cu98zrn

Figure 4.17: Research category sample tweets [Ward’s algorithm]

The treatment category has 4 different clusters. In addition, irrelevant tweets are clustered under the treatment categories (highlighted in figures 4.19, 4.20, 4.21 and 4.22).

Treatment
rt @lucykapasiitv: cancer fighter @riya_dandekar with her mum at #raceforlife bham. riya is 21 & needs a stem cell tplant. @itvcentral http
stem cell therapy could lead to hiv cure - sfgate http://t.co/e7siyd3mse via @sfgate
stem cell therapy is hampered by increased tumors from residual pluripotent cells after treatment b http://t.co/m31i1rpbkp
stem cell nutrition could eclipse antioxidant supplement market. #precisely. #health #stemcells http://t.co/ezt3kzrj8s @jjdulingint
ms stem cell therapy treatment hope for mum http://t.co/0ugfatridg
stem cell nutrition could eclipse antioxidant supplement market. http://t.co/r758xz1z1p #precisely. #health #stemcells @jjdulingint
bioheart announces world's first combination stem cell treatment #newssales http://t.co/nt7unfdpqm
doh chief, nancy clash over stem cell funding

Figure 4.18: First treatment category sample tweets [Ward's algorithm]

Treatment
rt @riaus: it's the last day of the exhibition stem cell stories - if you can make it to the science exchange today, come see it http://t.c
aging stem cells may cause failures in cell therapy http://t.co/diuzncqwwv
http://t.co/yeczi0f3qc
featured protocol: generation of induced neuronal cells by the single factor ascl1 http://t.co/ntogx2nvm9 +12 free stem cell protocols more!
please visit easton's stem cell therapy fund! http://t.co/31lokcsffk
we're under way: new stem cell operation could revolutionise treatment of knee injuries http://t.co/onuvyvmpj1
photo: stem-cell: reidavidson: i was doodling x-kids and liked how this sketch turned out so i inked it.... http://t.co/rqbaszn59p
scientists are 1 step closer to stem cell therapy for ms patients
bioheart announces world's first combination stem cell treatment http://t.co/fstg11mdzi

new stem cell operation could revolutionise treatment of knee injuries http://t.co/az6n0ltle4

Figure 4.19: Second treatment category sample tweets [Ward's algorithm]

Treatment
good morning, brand new to the uk. cutting edge stem cell technology. anti ageing creams & supplements. also opportunity to earn money dm
@todbellydance chris (todmorden) has a stem cell match register @anthonynolan sat 26 july walsden cricket club 12-5 #helpothers #curecancer
stem cell therapies hold great promise in the application of neurodegenerative.. http://t.co/c1yghkl7u0 http://t.co/ww69nqqhek
news: #stemcells a new stem cell operation could revolutionise treatment of knee injuries http://t.co/qpiqj4viux
epigenetic regulation of adult stem cell function http://t.co/gi0wusgsll
rt @bucktoj: .@lancstelegraph chris (todmorden) has a stem cell match. register @anthonynolan sat 26th july walsden cricket club 12-5 #help
have you seen the eurostemcell stem cell map of europe? reddstar's september public event is listed there as... http://t.co/t9wavinssb
genome editing goes hi-fi: innovative stem cell technique
rt @thewayofjay: #geekspeak 8yrs after a stem cell treatment to treat paralysis, a woman started growing a working nose on her spine. http:
summary : for breast cancer patients, the era of... http://t.co/n8dzqnjdje
bioheart announces world's first combination stem cell treatment http://t.co/0kpwn1bc1e #biotech

Figure 4.20: Third treatment category sample tweets [Ward's algorithm]

Treatment
public broadcaster nhk apologizes for stalking female stem cell researcher like papparazzis http://t.co/o3zuknol5s
nice to know there is a correlation between enrile's stem-cell powered immortality and naia toilets
rt @curealliance: dri-china team is the first formally approved by the government to perform stem cell / advanced cell therapies in... http
rt @ccentenomd: are pharma stem cell companies gearing up to grow and mass distribute shoddy cells? http://t.co/hf3jhvqhdm http://t.co/af1s
copd stem cell treatment success http://t.co/qlhzck2ydd

autism a stem cell treatment success http://t.co/uhq4rbax1a
new stem cell operation could revolutionise treatment of knee injuries http://t.co/lpbui5bqko

Figure 4.21: Fourth treatment category sample tweets [Ward's algorithm]

4.6.3. DBSCAN

DBSCAN failed to cluster large number of tweets for this experiment because of its inability to handle large document feature matrix. So, only 5000 tweets were considered for this experiment.

Data Set	Tweets	Features
Stem Cell	5000	4126

Table 4.10: Dataset used for DBSCAN algorithm

The DBSCAN's output depends upon eps and min_sample parameters [49]. To get accurate results, one has to choose the value of these factors carefully. Applying this algorithm on datasets, it classifies most of the tweets as noise and it is very sensitive to min_sample parameter because number of clusters keeps on increasing as the min_sample decreases as shown in tables below.

EPS	MIN_SAMPLE	CLUSTERS	NOISE
0.001	7	65	1227
0.001	10	48	1359
0.001	15	30	1567
0.001	20	21	1726
0.003	7	65	1227
0.003	10	48	1359

0.003	15	30	1567
0.003	20	21	1726
0.005	7	65	1227
0.005	10	48	1359
0.005	15	30	1567
0.005	20	21	1726

Table 4.11: Output of DBSCAN

Figure 4.23 shows the relationship of Min_Sample parameter with Noise. As Min_Sample size is increased, Noise increases gradually. Thus, it can be said that Min_Sample is directly propositional to Noise. Moreover, number of clusters keeps on increasing with decrease in value of Min_Sample. On the other hand, EPS does not have much impact on number of clusters and Noise.

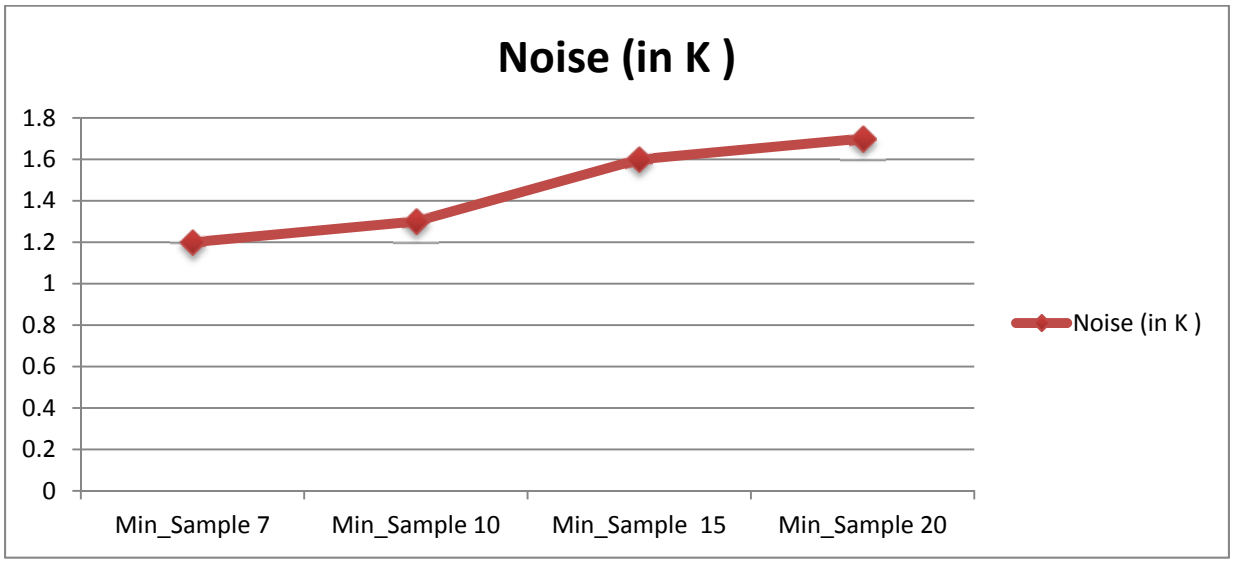


Figure 4.22: Min_Sample VS Noise

The DBSCAN clustered approximately one third of total tweets as noise, when results were run for *eps=0.001* and *min_sample=20*. Figure 4.24 shows some of the sample tweets from DBSCAN noise.

SAMPLE NOISE TWEETS
New stem cell operation could revolutionise treatment of knee injuries http://t.co/Ri922lEZLh via @guardian
Hair Regrowth Treatment Stem Cell Hair Restoration Technique: Hair Regrowth Treatment Stem Cell Hair Restorati... http://t.co/tLb8ZzxYKu
@Team Inquirer @leisalaverria THE LADY IS RIGHT. STEM CELL TREATMENT IS ONLY FOR THE RICH NOT THE POOR FILIPINOS.
Promising #Southampton #Abicus #stemcell #knee injury treatment regenerates damaged tissue http://t.co/vKs5fZ3uCp #sportsinjury
Regrowth Treatment Stem Cell Hair Restoration Technique http://t.co/TtPLgjGtHb
RT @VSinghhh: Meditation rather than medication: discussed Stem Cell research & new treatments. Consensus: we prefer prevention! http://t.c
RT @thewayofjay: #GeekSpeak 8yrs after a stem cell treatment to treat paralysis, a woman started growing a working NOSE on her spine
I would love to try stem cell treatment for my lower back pains. I hear it's pretty established in Germany. Anyone know anything about this?
New biomarkers found may lead to developing stem cell replacement treatments in the inner ear. #audiology #audpeeps http://t.co/WgDamYkfgv

Figure 4.23: DBSCAN sample noise tweets

These are the tweets related to treatment category, but DBSCAN has considered them as noise.

The experiments were also performed on tweets gathered based on "Obama" query from the Streaming Twitter API. The results are published in a conference paper [59].

5. CONCLUSIONS & FUTURE WORK

5.1. Summary

Document clustering is a promising and challenging research area. It can be used in a variety of data mining fields to help discover meaningful patterns of data. It divides the data set into groups of similar documents. In this thesis, clustering has been studied in order to perform semantic analysis on tweets. Clustering algorithms are generally of two types: Partition based clustering and Hierarchical clustering.

Partition based clustering methodologies use a defined number of clusters as an input parameter, then randomly select the centroid of the cluster and produce clusters of flat shape. As such, the accuracy of these algorithms depends primarily on the initial input parameter. The literature also shows that they do not work well on large datasets.

The hierarchical clustering techniques build clusters in the form of a tree by either merging or splitting them iteratively. This technique has two general categories: Agglomerative Hierarchical Clustering and Divisive Hierarchical Clustering. Agglomerative clustering forms clusters using a Bottom-Up approach, whereas Divisive Clustering builds clusters using a Top-Down approach. These techniques also need prior information about the number of clusters.

Density based algorithms are typically used more often in clustering. They are called density based because they cluster in areas of higher density and separate from areas of lower density. In comparison to k-means, clusters formed by density-based algorithms are of convex shape. This technique does not require prior information about the number of clusters, but produces clusters itself. However, there are two primary

inputs to the algorithms: *Min_Sample* and *EPS*, which define the density required to build the cluster.

5.2. Conclusion

The hypothesis was that the proposed methodology would preprocess tweets to remove the clutter, extract relevant features from each tweet and cluster tweets into meaningful categories. The similarity between tweets was to be determined using cosine similarity score, which analyzes the degree of relativity between the tweets. The motivation of the research was to develop a methodology that does not require a-priori information about the number of clusters or require any input parameters but rather forms clusters based on their determined similarity.

The proposed methodology dynamically creates categories and sub-categories as well as adjusts categories based on cosine-similarity. It has been shown to effectively categorize tweets gathered from the Twitter API.

After analyzing the related work in the area of document clustering, the combinatorial tweet clustering methodology proposed a combination of both agglomerative and divisive clustering approaches. It is a feature based clustering methodology that does not require any prior information about the number of clusters but rather creates clusters based on determined cosine similarity score. Cosine similarity score represents the degree of relativity between two n-dimensional vectors. It creates final clusters in three steps:

i. Select the top N features and create N broader clusters around these categories.

ii. *Divisive Clustering (Top-Down):* Calculate intra-cosine similarity score

for each broader cluster and divide the clusters according to a given threshold. The division continues until boundary condition is reached.

iii. *Agglomerative Clustering (Bottom-Up):* Calculate inter-cosine similarity score for each cluster from step ii. Merge most similar clusters and delete redundant tweets.

At the end of these steps final clusters and categories are created.

It has been shown that there are issues with DBSCAN, Ward's Hierarchical Clustering and k-means algorithms for appropriately categorizing tweets. The k-means algorithm is a heuristic algorithm that uses a random initiation of a centroid and as such, this random selection affects the output depending on what centroid is chosen. In addition, k-means and Ward's algorithm does not predict the number of categories emerging because the algorithms are initialized with the number of categories the data is to be divided into. This is a problem because tweets that are irrelevant appear in any given cluster.

DBSCAN can be reasonably accurate in clustering tweets but the issue is that it may consider a large number of tweets to be noise. In addition, one has to carefully choose the values

of *eps* and *min_sample* depending upon the dataset, which can significantly impact the accuracy of clustering.

The proposed methodology has addressed the items identified in the introduction of chapter 3. These items are stated again here:

Item 1: *The preprocessing of tweets to reduce dimensionality of features is important for accuracy because it may affect the results.*

Much of the literature points towards document pre-processing. Therefore, it is challenging to get concise feature set from Twitter data because tweets contain irrelevant information such as: informal language, URLs, usernames and the like that can significantly affect the accuracy and quality of results. The proposed methodology carefully pre-processes the tweets and remove features that contribute little in the clustering process.

Item 2: *When hierarchical clustering is applied as a classification tool, there are mistakes that can be made at early stages that may not be corrected at later stages.*

It has been identified that hierarchical clustering (Top-down or bottom-up) may lead to mistakes at early stages in the hierarchy which are very difficult to correct at later phases of clustering. Therefore, this methodology provides two-way traversing of the tree in order to potentially circumvent initial mistakes. If there is in-accurate clustering in the early steps, then the methodology revisits clusters and re-examines the similarity between the clusters.

Item 3: *Providing input parameters to clustering algorithms affects the accuracy and type of clusters.*

Input parameters such as the number of clusters, cluster density (MinPts), Eps and the like define the accuracy of the clustering process. To solve this problem, this methodology does not take any input parameters and is insensitive to the number of clusters required. The number of clusters are determined based on the dataset being used.

Item 4: *Similar tweets in each cluster may also be iteratively divided into sub-clusters, which can provide insights of knowledge more efficiently.*

It has been determined that even after creating clusters and their categories that may still be sub-divided iteratively into sub-categories or sub-clusters to arrange the data more efficiently. This methodology provides sub-categorization of clusters and divides tweet categories so that they can be browsed further and made more meaningful.

The contributions of this thesis are:

- A combinatorial clustering methodology that creates categories dynamically without requiring any initial input parameters.
- A combinatorial clustering methodology that has the ability to divide categories incrementally into sub-categories for improved clustering.
- A combinatorial clustering methodology that shows an increase in clustering quality and effectiveness as compared to standard algorithms.

5.3. Future Work

Some future work for this research is explained in the following sections:

5.3.1. Feature Extraction

Basic feature extraction techniques have been used in this research. There are also a number of preprocessing techniques that could be developed to produce a more precise feature set. A better term weighing approach could be investigated to accurately identify the most important features. Timed relationships could also be taken into consideration using POS tags other than nouns.

Further work can also be done to attempt to infer meaning from feature sets within individual tweets based on determined feature patterns. This work can be extended further to analyze the synonym relationship between words and bi-grams. In addition, pre-processing techniques could be developed to detect casual language words and modify them to their original form.

5.3.2. Clustering

This clustering methodology could be modified further to reduce computational time. Some other factors in tweets could also be taken into consideration such as User Information, RT, hash-tags and time-stamps. Using different distance measuring techniques could also extend this work further. This data mining approach has been performed on Twitter, but could also be extended further for Facebook and other social media.

5.3.3. Text Representation

The Vector Space Model (VSM) representation has been employed in this work. However, it has some limitations such as:

i. High Dimensionality

ii. Loss of correlation with neighbors

iii. Loss of semantic relationship between words

Semantic relationship between words could be preserved using Ontology models. Ontology models help keep the domain knowledge of a term in a given document. Another document representation scheme is Latent Sematic Indexing (LSI), which could also be used to preserve representative features in a document.

To preserve the correlation between terms, the Universal Networking Language

(UNL) technique could be used. It represents the document as a graph where nodes represent terms and links represent the relationship between them [50].

LIST OF REFERENCES

[1] A. Costill. (2013, Dec 10). *25 Insane Social Media Facts* [Online]. Available: http://www.searchenginejournal.com/25-insane-social-media-facts/79645/. Accessed on: Dec 2, 2014.

[2] B. Allie and O. Merver, "Methods Advertisers Use on Social Media Sites", *Student Journal of Media Literacy Education*, Vol. 1, No. 1, 2010.

[3] Twitter Inc. (2014). *About Twitter* [Online]. Available: https://about.twitter.com/company. Accessed on: Dec 2, 2014.

[4] C. C. Aggarwal and C. X. Zhai, *Chapter 4: A survey of text clustering algorithms* [Online]. Available: http://www.charuaggarwal.net/text-cluster.pdf. Accesses on: Dec 5, 2014.

[5] J. Weissbock et al., "Using External Information for Classifying Tweets" in *Brazilian Conference on Classifying Tweets*, pp 1-5, 2013.

[6] Y. Cheng et.al, "A Document Clustering Technique Based on Term Clustering and Association Rules", *IEEE,* 2010.

[7] A.K. Jain et al. "Data Clustering: A Review" in *ACM Computing Surveys (CSUR),* Vol. 31, No. 3, pp. 264-323, 1999.

[8] P. Willett, "Recent trends in hierarchic document clustering: a critical review", *Information Processing and Management*, Vol. 24, No. 5, pp. 557-597, 1988.

[9] E. M. Voorhees, "Implementing Agglomerative Hierarchic Clustering Algorithms for use in document retrieval", *Information Processing & Management*, Vol. 22, No. 6, pp. 465-476, 1986.

[10] N. Jardine and C. J. Van Rijsbergen, 1971. "The use of hierarchic Clustering in information retrieval". *Information Stor. Retr.,*Vol. 7, pp 217-240, Great Britain, 1971.

[11] D.R. Cutting et al., "Scatter/gather: A cluster-based approach to browsing large document collections" in *International Conference on Research and Development in Information Retrieval*, pp. 318-329, Copenhagen, Denmark, SIGIR, 1992.

[12] L. Kaufman and P. J. Rousseeuw, "Finding groups in data: An introduction to cluster analysis" in *New York: John Wiley & Sons, Inc.,* March 1990.

[13] B. Larsen, & C. Aone, "Fast and effective text mining using linear-time document clustering" in *International Conference on Knowledge Discovery and Data Mining*, pp. 16-22, San Diego, California, United States, 1999.

[14] Scikit Learn (2014), *2.3. Clustering* [Online], Available: http://scikit-learn.org/stable/modules/clustering.html. Accessed on: Dec 5, 2014.

[15] M. Steinbach et al., *A Comparison of Document Clustering Techniques*, University of Minnesota, Minnesota - 55455. Available: http://www.cs.cmu.edu/~dunja/KDDpapers/Steinbach_IR.pdf. Accessed on: Dec 6, 2014.

[16] J. Wang, "Encyclopedia of Data Warehousing and Mining", USA, UK: Idea Group Reference, 2006, pp. 555-559.

[17] M. Steinbach et al., “A comparison of document clustering techniques” in *Workshop on Text Mining*, SIGKDD’00, 2000.

[18] T. Velmurugan and T. Santhaman, “A Comparative Analysis between K-medoids and Fuzzy C-Means Clustering Algorithms for Statistically Distributed Data Points”, *Journal of Theoretical and Applied Information Technology,* Vol. 27, No. 1, May 15, 2011.

[19] L. Kaufman and P.J. Rousseeuw, “Finding groups in data: An introduction to cluster analysis”, *New York: John Wiley & Sons, Inc*., March 1990.

[20] R. Krishnapuram et al., “A fuzzy relative of the k-medoids algorithm with application to document and snippet clustering” in *IEEE International Conference - Fuzzy Systems, FUZZIEEE 99*, Korea, August, 1999.

[21] M. Ester et al., “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise” in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[22] H. Backlund et al., “DBSCAN: A Density-Based Spatial Clustering of Application with Noise”, Linkoping University – ITN, Rep. TNM033, Nov. 30, 2011.

[23] K. Wang et al., “Clustering transactions using large items” in *International Conference on Information and Knowledge Management*, CIKM’99, Kansas City, Missouri, United States, pp. 483-490, 1999.

[24] F. Beil et al. “Frequent term- based text clustering” in *International Conference on Knowledge Discovery and Data Mining*, KDD’02, Edmonton, Alberta, Canada, pp. 436-442, 2002.

[25] B. Fung et al., “Hierarchical document clustering using frequent itemsets SIAM” in *International Conference on Data Mining*, SDM’03, San Francisco, CA, United States, pp. 59-70, 2003.

[26] D. Sculley, “Web-Scale K-Means Clustering”, *WWW*, Raleigh, North Carolina, USA, April 2010.

[27] A. Mihael, “OPTICS: Ordering Points to identify the clustering structure” in *Int. Conf. on Management of Data*, Philadelphia PA, 1999.

[28] Xu Xiaowei et al., “A Distribution Based Clustering Algorithm for Mining in Large Spatial Databases” in *Proceedings of 14th Int. Conf. on Data Engineering*, 1998.

[29] B. Rupanka et al., “A Survey of Some Density Based Clustering Techniques” in *National Conference on Advancements in Information, Computer and Communication – AICC’13,* Jaipur, Rajasthan, India, March 2013.

[30] A. Hinneburg and A. D. Keim, “An Efficient Approach to Clustering in Large Multimedia Databases with Noise”, *American Association for Artificial Intelligence,* 1998.

[31] K. Sungchul et al., “Finding Core Topics: Topic Extraction with Clustering on Tweet” in *2nd Int. Conf. on Cloud Computing and Green Computing,* pp. 777-781, 2012.

[32] A. Antenucci et al., "Classification of tweets via clustering of hash tags", Final Project, pp 1-11, 2011.

[33] M.D. Bhoomija, "Comparison of Partition based clustering algorithms", *Journal of Computer Applications*, Vol. 1, No. 4, pp. 18-21, 2008.

[34] Clustering, *Chapter 3: Cluster Analysis* [Online], Available: http://www.inf.unibz.it/dis/teaching/DWDM/slides2010/lesson8-Clustering.pdf. Accessed on: Dec 9, 2014.

[35] R. Ng and J. Han, "Efficient and effective clustering method for spatial data mining" in *VLDB,* 1994.

[36] A.K. Patidar et al., "Analysis of different similarity measure functions and their impacts on shared neighbor clustering approach" in *Int. journal of computer application (0975-8887),* Vol. 40, No. 16, 2012.

[37] Anna Huang, "Similarity Measures for Text Document Clustering" in *NZCSRSC'08*, Christchurch, New Zealand, April 2008.

[38] K. Taghva and R. Veni, "Effects of Similarity Metrics on Document Clustering" in *Seventh International Conference on Information Technology*, 2010.

[39] A. Moon, "A Survey on Document Clustering with Similarity Measures" in *Int Journal of Advanced Research in Computer Science and Software Engg.*, Vol 3, No. 11, pp. 559-601, 2013.

[40] *Mining Similarity Using Euclidean Distance, Pearson Correlation, and Filtering* [Online], Available: http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/similarity.html. Accessed on: Dec 17, 2014.

[41] K. Kouser and Sunita, "A comparative study of K Means Algorithm by Different Distance Measures", *International Journal of Innovative Research in Computer and Communication Engineering,* Vol. 1, No. 9, pp. 2443-2447, 2007.

[42] A. Amine. et al., "Evaluation of Text Clustering Methods Using WordNet", *The International Arab Journal of Information Technology*, Vol. 7, No. 4, pp. 349-355, 2010.

[43] Steven Bird et al. (Nov 3, 2014), "Language Processing and Python" from *Natural Language Processing with Python* [Online], Available: http://www.nltk.org/book/ch01.html#fig-fdist-moby. Accessed on: Dec 18, 2014.

[44] *Alphabetical list of part-of-speech tags used in the Penn Treebank Project* [Online] Available: http://www.ling.upenn.edu/courses/Fall_2007/ling001/penn_treebank_pos.html. Accessed on: Dec 18, 2014.

[45] P.D. Turney and P. Patrick, "From Frequency to Meaning: Vector Space Models of Semantics", *Journal of Artificial Intelligence Research*, Vol. 37, pp. 141-188, 2010.

[46] A. Huang, "Similarity Measures for Text Documents Clustering" in *NZCSRSC*, Christchurch, NZ, 2008.

[47] Olga Kharif (May 24, 2012), *'Likejacking': Spammers Hit Social Media* [Online], Available: http://www.businessweek.com/articles/2012-05-24/likejacking-spammers-hit-social-media. Accessed on Dec 20, 2014.

[48] C.S. Perone (Sep 12, 2013), *Machine Learning: Cosine Similarity for Vector Space Models (Part III)* [Online], Available: http://pyevolve.sourceforge.net/wordpress/?p=2497. Accessed on: Dec 20, 2014.

[49] M. Ester et al., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise" in *KDD*, 1996.

[50] B.S. Harish et al., "Representation and Classification of Text Documents: A Brief Review", *Recent Trends in Image Processing and Pattern Recognition,* pp. 110-115, 2010.

[51] E.E. Milios et al., "A Systematic Study on Document Representation and Dimensionality Reduction for Text Clustering", Dalhousie University, Halifax, Nova Scotia, Tech. Rep. CA CS-2006-05, July 2006.

[52] A. Olowe et al., "A Survey of Data Mining Techniques for Social Media Analysis" in *Journal of Data Mining & Digital Humanities,* June 2014.

[53] Wikipedia (April 7, 2015), *k-means clustering* [Online], Available: http://en.wikipedia.org/wiki/K-means_clustering. Accessed on: March 17, 2015.

[54] B. S. Everitt et al., "Cluster Analysis", *Oxford University Press*, fourth edition, 2001.

[55] Steven Bird et al. (March 16, 2015), *Accessing Text Corpora and Lexical Resources* [Online], Available: http://www.nltk.org/book/ch02.html, Accessed on: March 14, 2015.

[56] NLTK (Natural Language Tool Kit) Tokenization and Tagging [Online], Available: http://www.bogotobogo.com/python/NLTK/tokenization_tagging_NLTK.php, Accessed on March 25, 2015.

[57] Wikipedia (June 8, 2015), *tf-idf* [Online], Available: https://en.wikipedia.org/wiki/Tf%E2%80%93idf. Accessed on: July 14, 2015.

[58] Quora (April 24, 2012), *Is Cosine Similarity Effective?* [Online], Available: http://www.quora.com/Is-cosine-similarity-effective. Accessed on: July 14, 2015.

[59] Navneet K. & Craig M. Gelowitz, "A Tweet Grouping Methodology Utilizing Inter and Intra Cosine Similarity", *IEEE 28th Canadian Conference on Electrical and Computer Engineering",* Halifax, Canada, May 3-6, 2015.