International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 6 (Nov-Dec 2016), PP. 19-25

PATTERN DISCOVERY AND DOCUMENT CLUSTERING USING K-MEANS, PAM AND HAC

[1]Abdul Ameer Hussain[1], [2]Prof. Prajna Bodapati

[1]M.Tech-Student, [2]Professor
[1, 2] Dept. Computer Science and System Engineering
[1,2]Andhra university College of Engineering
Visakhapatnam, India
[1]ameercse55@gmail.com

***Abstract*— People search for important information which they are prone to use internet, but now a day's most of the information is stored in text such as in news articles, E-books, email message, blogs and web pages. This is very difficult to get accurate data what exactly people want. To make easier them we have to apply text mining process for pattern finding and clustering similar information from the text. In this paper we mining the frequent terms from documents and generating the plot diagram for frequent terms using RStudio IDE. We use Euclidean and cosine similarity methods. The aim of this paper is clustering the documents using k-means, PAM and HAC methods in R as statistical analysis tool and calculating the precision, recall and F-measure values for clusters and we compare the three clustering algorithms. For comparing we take the class labels documents from the 20 news group data set.**

***Keywords*— Text mining, stop words, stemming, TF – IDF, Clustering, k-means, PAM, HAC**

1. INTRODUCTION

Due to the heavy usage of electronics devices storing of information is a rapidly growing, mostly amount of information is available in electronic formats such as online newspapers, journals, text documents, pdf etc. Using all these electronic information, indexing text, controlling of data or easy way of searching is not feasible especially for search engines [2]. Thus, text mining that is to discover interesting patterns from large amount of text data within limited sources has become popular. Automatically Clustering of documents is most important in data mining task. Document clustering groups similar documents into one cluster, dissimilar documents into different cluster. Clustering of documents plays a vital role in effective document organization, pattern discovery and information extracting from documents [1]. Organizing the results returned by the search engine and in browsing documents clustering comes in hand. It also helps in identifying and focusing on the relevant set of results. Document clustering methods helps to insight into data distribution or preprocess data for applications. For example, if any search engine by using of clustered documents in order to search an item or object, it can produce results more effectively and efficiently.

Documents are generally in big volumes, high dimension and with complex semantics which are challenging problems of document clustering. Our motive in this present paper is to extract particular domain of work from a huge collection of documents using popular document clustering methods. Agglomerative hierarchical clustering, K-means and PAM are three clustering techniques that are commonly used for document clustering. K-means is used because of its efficiency, PAM is used as because of it over come disadvantage of k-means, and HAC is used because of its quality [13]. Document clustering is a more specific technique for document organization, RStudio tool has the automatic text mining process packages by using such packages we easily load the documents into RStudio environment. By using "tm" package we easily pre processed the documents i.e. removing of stop words, stemming of words etc[8]. Here we use the RStudio tool.

In this paper, we aim to cluster documents into clusters by using above three clustering methods and make a comparison between them. The comparison is done with the precision, recall and F-measure values for each and every clustering object. Here we take the sample 20news groups data set documents with the different class labels. Based on the class labeled documents we easily find out the how many clusters can be formed. This is helpful to find out the precision, recall and F-measure values. We perform the clustering by using k-means, PAM and HAC algorithms and compare the precision, recall and F-measure values for different size documents i.e. 40 documents with 2 class labels or 100 documents with 5 class labels etc.

This paper is organized as follows- Second section describes the literature survey in this area. Third Section describes the proposed methodology. Document clustering methods are described in fourth section. Fifth Section explains experimental evaluation and discussions. The paper is concluded in sixth section.

2. LITERATURE SURVEY

There are many researchers are doing paper on document clustering, present data mining concepts are implementing on

RStudio. The Steinbach and et al. worked on comparison of document clustering methods in 2000[7]. In their paper work applied Hierarchical clustering algorithms, k-means etc. The Delany et. al. [6] has done his paper work on the text mining technique using spam sms data set. In his paper R statistical analysis tool has been used to form the cluster of spam messages and for finding associations among words of messages. The Andrew et. al. has applied text mining on "Complete Works of William Shakespeare" stories data set downloaded from [11]. This work shows the how to find patterns from the text data sets. Here I follow the webpage "Basics of text mining process using r language" [18] how to clusters the documents using dtm matrix. This web site helps me "how to preprocessing the documents". The web page [14] had done the text mining process by using R statistical analysis tool on twitter data set. This helps to frequent terms plot diagram generating etc. There are some research works [7,8,9,10,12,13,15,16,17] shows examples on text mining which provides function for text mining like, stop words removal, stemming, whitespace removal, TF - IDF calculations, clustering etc.

3. PROPOSED METHODOLOGY

Text mining is process of dealing with unstructured data, a gigantic amount of data presence in the text documents. In a document all words all not important for text analysis. Approximately 2% words of document corpus is used for clustering and pattern analysis and remaining words are like stop words, numbers, white spaces etc. So we have to apply preprocessing task before frequent patterns generating and document clustering. After pre processing task completed, our corpus has only important words. The step by step process is explained by using figure 3.1 block diagram of proposed system.

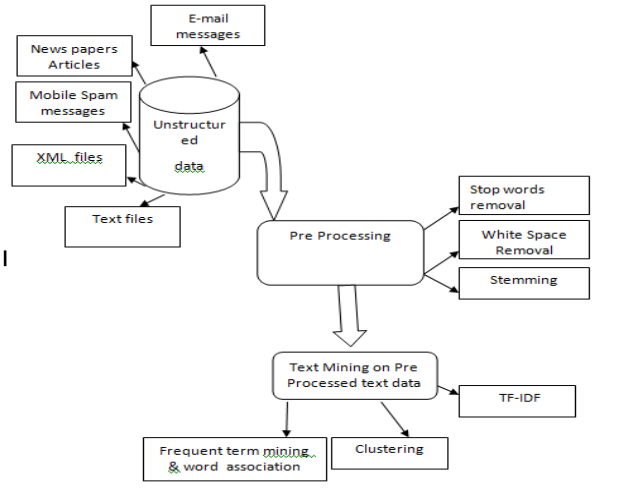

Figure 3.1. Block diagram

3.1 Preprocessing

Preprocessing is the major task in text mining process. Here we are filtering the lots of unwanted terms from the bag of words. For pattern finding tasks each document is treated as bag of words- as set of all words with the frequency of the word occurred in that document. In preprocessing task we remove unwanted terms from the corpus. Here we have some cleaning methods to preprocessing our corpus. Some documents have the implicit structure terms like titles, sections, paragraphs etc.

Step by step of preprocessing task is as follows.

Step 1: Convert all upper case letters into lower case letters example: "GAME" is converted into "game" here all upper case letters are changed into lower case letters.

Step 2: Removing of stop words, here we remove the stop words like "the, a, of etc" such types of words are unused for text analysis. We can remove some topic based keywords from documents like "Document" is term which is often used in documents clustering if you want to clusters document clustering journal papers the term "document" unnecessary for frequent pattern generation and clusters. So we can remove "document" keyword from corpus.

Step 3: Stemming process, this is most important task in preprocessing why because most of words have different spelling but meaning of word is similar .i.e. studying, studies, studied. Such type of words transforms to root words. For stemming process we use the Porter's Stemmer algorithm.

Example illustrating stemming process:

-ATIONAL - >ATE	relational-relate
-TIONAl- >TION	conditional-condition
-ENCI - >ENCE	valenci- valence
-ANCI - >ANCE	hesitanci- hesitance

Step 4: In this step we remove the punctuation marks like eg. , . ? etc) (

Step 5: Remove the numerical data from the documents, which are not required for pattern discovery.

Step 6: After completion of above 5 steps some white spaces are generated such type of white spaces are removed from documents.

After completion of preprocessing task our document corpus has the only useful words for text analysis.

3.2. Document Term Matrix

A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. In term document matrix rows are terms of corpus and columns are documents in corpus. In this document term matrix each term will be taken as column filed if document have that term it will be 1 otherwise 0. Figure 3.2 shows the document term matrix for 5 documents and 11 terms.

	Terms										
Docs	also	appl	banana	cost	day	fruit	grape	high	like	orang	veget
1	0	1	1	0	0	0	0	0	1	0	0
2	0	1	1	1	1	0	0	1	0	0	0
3	1	0	0	1	0	0	1	1	0	1	0
4	0	1	1	0	0	1	1	0	0	1	0
5	1	0	0	1	1	0	0	1	0	0	1

Figure 3.2. Document terms matrix

3.3. TF-IDF

This step involves the calculating weight of term. Tf-idf stands for term frequency-inverse document frequency. The text mining and information retrieval process often use tf-idf weight[23]. This weight is a statistical measure used to evaluate how important a term is to a document in a collection or corpus. The importance of word increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. The tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. Tf-idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

TF: Term Frequency, measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is mostly divided by the total number of terms in the document as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing

IDF(t) = log e(Total number of documents / Number of documents with term t in it)

TF - IDF weight=TF(t) * IDF(t)

For text matching, the attribute vectors A and B are usually the tf vectors of the documents. The cosine similarity can be seen as a method of normalizing document length during comparison.

3.3. Similarity Measures

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of similarity or dissimilarity of the target documents and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. All clustering methods have to assume some cluster relationship among the data objects that they are applied on. Similarity between a pair of objects can be defined either explicitly or implicitly. The nature of similarity measure plays a very important role in the success or failure of a clustering method. Some of the similarity measures explained briefly below based on single view point and multi view point.

A. Euclidean Distance

Euclidean distance is a regular metric for geometrical problems. It is the common distance between two points and can be without difficulty measured with a ruler in two- or three dimensional space. It is also the default distance measure used with the K-means algorithm. Euclidean distance is one of the most popular measures: Dist(di, dj) = ||di − dj ||. It is used in the traditional k-means algorithm. The objective of k-means is to minimize the Euclidean distance between objects of a cluster and that clusters centroid. Measuring distance between text documents, given two documents da and db represented by their term vectors ta and tb respectively, the Euclidean distance of the two documents is defined as

$$D_E(\overrightarrow{t_a}, \overrightarrow{t_b}) = \left(\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2\right)^{1/2},$$

Where the term set is T = {t1. . . tm}. We use the tf-idf value as term weights

B. Cosine Similarity

Cosine similarity is a measure of similarity between two vectors of *n* dimensions by finding the cosine of the angle between them, often used to compare documents in text mining. Given two vectors of attributes, *A* and *B*, the cosine similarity, θ, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}.$$

3.5 Document Clustering Methods

3.5.1 K-Means Method

The k-means clustering algorithm is known to be efficient in clustering large data sets. It aims to partition a set of documents, based on their similarities between documents. The main idea is to define *k* centroid, one for each cluster. The centroid of a cluster is formed in such a way of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance to all documents in that cluster.

K- Means algorithm:

Input: DocSet= {d1,d2, …,dn}, Set of documents binding as Corpus

Steps:

1. Select K document vectors as the initial centroids of K clusters.
2. Repeat
   For i=1,2,…, n
3. Compute similarities between di and K centroids.
4. Put di in the closest cluster

5. END For
6. Recomputed the centroid of the cluster until the centroid doesn't change.

Output: K clusters of Text Documents

3.5.2 Partition Around Medoids (PAM) Method

Both the *k*-means and PAM algorithms are partition (breaking the dataset up into groups). *k*-medoids chooses data points as centers (medoids or exemplars).*K*-medoids is also a partitioning technique of clustering that clusters the data set of *n* documents into *k* clusters with *k* known medoids.

PAM algorithm:

Input: DocSet= {d1,d2, …,dn}, Set of documents binding as Corpus

Steps:

1. Initialize: select [citation needed] k of the n data points as the medoids.
2. Associate each data point to the closest medoid
3. While the cost of the configuration decreases:

I. For each medoid *m*, for each non-medoid data point *o*:

i. Swap *m* and *o*, recomputed the cost (sum of distances of points to their medoid)
ii. If the total cost of the configuration increased in the previous step, undo the swap

Output: K clusters of Text Documents

3.5.3 *Hierarchical Agglomerative Clustering (HAC) Method*

Hierarchical clustering clusters similar instances in a group by use of a similarity (distance) measure which is generally Euclidean measure in general, and cosine similarity for documents. Hierarchical clustering can be categorized into two; agglomerative (bottom-up) and divisive (top-down) clustering. An agglomerative clustering algorithm starts with clusters which each of them contain only one instance and for each iteration merges the most similar clusters until the stopping criterion is met such as a requested number *k* of clusters is achieved. A hierarchical clustering is often represented as a dendrogram.

HAC algorithm:

Input: D= {d1,d2,…, dn},Corpus of documents

d1= {t11, t12,…, t1n}, d2={t21, t22,…, t2n}…documents containing terms.

Steps:

1. Compute the similarity between all pairs of clusters, i.e., calculate a similarity matrix whose ijth entry gives the similarity between the ith and jth clusters.

Cos(d1,d2)=d1.d2/|d1|X|d2|

d1.d2=(t11*t21+t12*t22+…+t1n*t2n)

$|d1|=\sqrt{[(t11)^2+(t12)^2+\ldots+(t1n)^2]}$

$|d2|=\sqrt{[(t21)^2+(t22)^2+\ldots+(t2n)^2]}$

2. Merge the most similar (closest) two clusters.
3. Update the similarity matrix to reflect the pair wise similarity between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.

Output: Dendogram of Documents clusters

3.6 *Testing*

To evaluate the clustering results, precision, recall, and F-measure were calculated over pairs of points.

Correct decision:

TP = Decision to assign two similar documents to the same cluster.

TN = Decision to assign two dissimilar documents to different clusters.

Error:

FP = Decision to assign two dissimilar documents to the same cluster.

FN = Decision to assign two similar documents to different clusters.

TABLE 3.1: Confusion matrix

	Same	*Different*
Same	TP	FN
Different	FP	TN

*Precision***:** Precision is the fraction of retrieved documents that are relevant to the search. It takes all the retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. In a text search on a set of documents precision is the number of correct results divided by the number of all returned results.

$$precision = \frac{TP}{TP + FP}$$

Recall: Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}.$$

F-measure: F-measure is the harmonic mean of precision and recall.

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

4. Experimental Evaluation

This section presents the experimental evaluation of three popular documents clustering algorithms k-means, PAM, HAC. The RStudio toolkit has been used to generate the

frequent terms finding, patterns clouds formations and clustering performing. In R statistical analysis tool has some packages by using those libraries we can easily perform the text mining process in R studio.

4.1 *Data Set*

For the performance evaluation, Newsgroups data set documents are selected. We use the 2 class labels documents which are 5 documents from the atheism categories and 5 documents from the baseball category.

TABLE 4.1: Sample 10 documents

Sino	*Document Name*	*Text*
1	Doc.athm1	I found a list of Biblical contradictions and cleaned it up a bit, but now I'd like some help with it. And does anyone know who originally wrote this list?
2	Doc.athm2	I didn't have time to read the rest of the posting, but I had to respond to this. I am absolutely NOT a "Messianic Jew". Another mistake. Sorry, I should have read alt. messianic more carefully.
3	Doc.athm3	This (frayed) thread has turned into a patented alt. atheism 5-on-1 ping-pong game, and I don't have any strong disagreement, so I'll try to stick to the one thing I don't quite follow about the argument It seems to me that there is a contradiction in arguing that the Bible was "enlightened for its times" (i.e. closer to what we would consider morally good based on our standards and past experience) on the one hand [I hope this summarizes this argument adequately], and on the other hand
4	Doc.athm4	As for rape, surely there the burden of guilt is solely on the rapist? Unless you force someone to live with the rapist against his will, in which case part of the responsibility is yours. I'm sorry, but I can't accept that. Unless the rapist was hypnotized or something, I view him as solely responsible for his actions. Not necessarily, especially if the rapist is known as such. For instance, if you intentionally stick your finger into a loaded mousetrap and get snapped, whose fault is it?
5	Doc.athm5	Neither was he a lunatic. Would more than an entire nation be drawn to someone who was crazy. Find an encyclopedia. Volume H. Now look up Hitler, Adolf. He had many more people than just Germans enamored with him.
6	Doc.baseball1	Yeah, Morris just knows how to win. That's why he lost 18 for Detroit in 1990. Funny how he wins a lot of games when he pitches on good teams but loses a lot when he pitches on bad ones. And if "rings" was the only criteria for success, then teams would always tend to repeat, and eventually you'd have the same team win the WS every bleep in year. Sort of like the yanks in the 50s. Morris is a decent pitcher on the downside of a good, not great, career. Toronto will finish 3rd or 4th this year, with Morris and all those rings, because their pitching staff was destroyed over the off-season.
7	Doc.baseball2	Last night, Boston Red Sox win its 11 games of 14 games by beating Seattle 5-2. Roger Clemson pitch not so dominate. He walked at least 6 man in first 6 inns. But Valetin and Greenwell hit homeruns and Red Sox prevail. I think that game is must win for Red Sox in Seattle, considering Darwin will faced Seattle ace Randy Johnson tonight.
8	Doc.baseball3	Where did Acker get a ring from? I would have to say that they are about even. I believe Acker got a ring from his wife when they were married the Blue Jays had such a strong offense? Don't tell me that Morris has this magical ability to cause the offensive players to score more runs. I don't know why you guys keep bickering about Morris. The stats show he is a mediocre pitcher at best (this year is another case), he just happened to win 21 games. I saw many of his games last year, he did pitch some good games. But this crap about being a clutch pitcher is nonsense, he was constantly giving up go ahead runs in the 6-8th innings (the clutch innings) and the Jays would somehow scrape a win for him. Another major factor in his 21 wins, is that Cito 'I don't realize i have a bullpen' Gaston would leave Morris in forever, therefore giving him many more chances to win games (i believe this is the major reason he won 21 games last year).
9	Doc.baseball4	The best one I saw last year was Willie McGee off Matthews (I think?) in Phillie. A fierce line drive that was still rising when it hit the second deck facade at the Vet. Willie McGee had one homerun last year.

10	Doc. baseball5	Most of tirade deleted .. I have an editor and know how to use i Okay we've been conservative and added about 18 wins so far. Now we're adding about 4 more wins thanks to the expansion teams. Okay, that's 22 wins. Lesse dipshit math genius, 72 + 22 = 94 . I think that's good enough to win the worse division in baseball? Next time, before you say something foolish, get a clue first! Either this is an example of *great* sarcasm or I'm really, really worried.

4.2 Performance Evaluation Measure

Before performing the clustering first of all do the preprocessing of documents we convert the upper case letters into lower case letters, remove the stop words, stemming the documents, removing of numbers , removing of punctuations, and stripping the white spaces from documents. Second step is to calculate the DTM for the document which is shown in figure 5.1; from DTM matrix we find frequent patterns and form word clouds, plot diagrams. Third step is to remove the sparse terms from the DTM. Fourth step is to find out the TF-IDF values matrix for the removed sparse terms from DTM matrix. Fifth step is to find out the distance matrix for TF-IDF matrix. Here we use the Euclidean and cosine distance methods. The step wise output results, figures and values shown below.

```
> dtm.sr <- DocumentTermMatrix(bthm.bab.cl,control=list(minWordLength=2,minDocFreq=2))
> dtm.sr
<<DocumentTermMatrix      (documents: 10, terms: 211)>>
Non-/sparse entries      : 235/1875
Sparsity                 : 89%
Maximal term length      : 12
Weighting                : term frequency (tf)
```

Figure 4.1: documents terms matrix

Frequent terms occurred in document term matrix shows the word clouds figure 4.2 is maximum 10 times present in corpus. Figure 4.3 shows the frequently 5 times occurred terms in Document Term Matrix. Sixth step is to apply the clustering methods on distance matrix figures are shown below

```
> wordcloud(names(freq), freq, max.words=10, rot.per=0.2, colors=dark2)
```

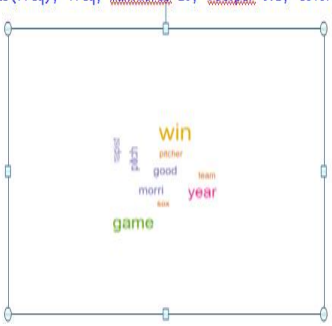

Figure 4.2: word cloud for maximum words 10

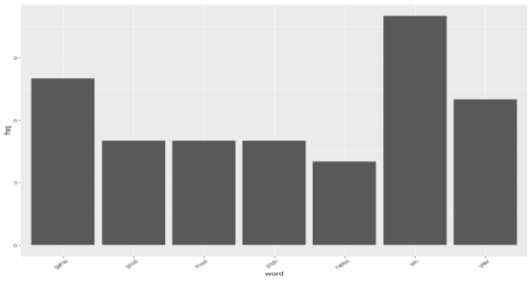

Figure 4.3: maximum 5 times occurred terms from DTM

4.3 Clustering results

Below figures shows the three clustering techniques applied on documents and their results.

```
K-Means results:
> table(k.cl$cluster)

 1  2
 5  5

> k.cl$cluster
 1  2  3  4  5  6  7  8  9 10
 2  2  2  2  2  1  1  1  1  1
```

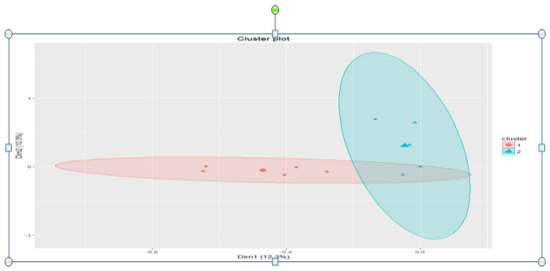

Figure 4.4: k –means result and plot diagram

```
> hc.avg
Call:
      Hclust (d = dtm.sr.cdis, method = "average")

      Cluster method          : average
      Distance                : cosine
      Number of objects       : 211
```

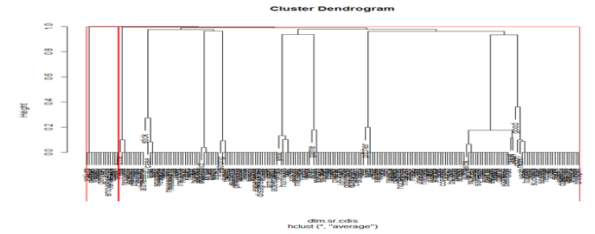

Figure 4.5: HAC result

```
PAM result:
> pam.res$cluster
 1  2  3  4  5  6  7  8  9 10
 1  2  1  1  2  2  2  2  2  2
> table(pam.res$clustering)

 1  2
 3  7
```

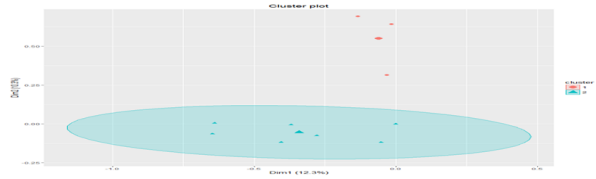

Figure 4.6: PAM clustering method

4.3 Precision, Recall, and F-measure

In this section we calculate the precision, recall and F-measure values for document clustering methods. We take different size of documents with class labels below 3 tables shows the precision, recall and F-measure values.

TABLE 4.2: K-Means Values

Algorithm	K –Means		
Documents	Precision	Recall	F-measure
10doc	1.0	0.95	0.97
40 doc	0.89	0.71	0.78
100 doc	0.55	0.35	0.42

TABLE 4.3: PAM Values

Algorithm	PAM		
Documents	Precision	Recall	F-measure
10doc	1.0	0.60	0.75
40 doc	0.86	1.0	0.93
100 doc	0.58	0.41	0.48

TABLE 4.4: HAC Values

Algorithm	HAC		
Documents	Precision	Recall	F-measure
10doc	0.60	0.50	0.54
40 doc	0.65	0.75	0.69
100 doc	0.49	0.27	0.34

5. CONCLUSION

R as statistical analysis tool might be helpful for preprocessing, weight etc types of work associated with text-mining. This project presented the results of an experimental study of some common document clustering techniques. In particular, we compared the three main approaches to document clustering, agglomerative hierarchical clustering, PAM and K-means. Our results indicate that the PAM technique is better than the standard K-means approach and as good as or better than the hierarchical approaches that we have tested. More specifically, the PAM approach produces significantly better clustering solutions quite consistently according to the F-measure and overall similarity measures of cluster quality. Further, we are trying to implement the various probabilistic and statistical models for selection of feature vectors from text corpus before applying text mining using RStudio IDE. We are trying to implement all text mining tasks for Retures2158 documents data set and opinion, review data.

REFERENCES

[1] G.K.Gupta, Introduction to Data Mining with Case Studies, PHI 2006,

[2] Han, J. and Kamber, M., *Data Mining: Concepts and Techniques, 2nd ed*., Morgan Kaufmann Publishers, 2006.

[3] Vikram Paudi, P. Radha Krishna,Data Mining,Oxford University Press, First Edition, 2009

[4] *Jiawei Han, Micheline Kamber, Jian Pei, DATA MINING Concepts and Techniques, Elsevier, Third Edition, 2012*

[5] *Arun K Pujari, DATA MINING TECHNIQUES, Universities Press, Second Edition, 2009*

[6] *S J. Delany, M. Buckley & D. Greene (2012) "SMS spam filtering: methods and data" Expert Systems With Applications 39, p 9899-9908, http://www.elsevier.com/*

[7] M.K.V.Anvesh and Dr.B.Prajna "Potential based similarity metrics for implementing hierarchical clustering", 103-volume-4-issue ,IJECS ,2015

[8] http://michael.hahsler.net/SMU/7337/install/tm.R

[9] http://cran.r-roject.org/web/packages/tm/vignettes/tm.pdf

[10] B.Prajna , Shashi M , "Document Clustering Technique based on Noun Hypernyms", IJECT Vol. 2, SP-1, Dec. 2011.

[11] http://faculty.washington.edu/jwilker/tft/Stewart.LabHandout.pdf

[12] Andrew, Text Mining the Complete Works of William Shakespeare, R-blogs, Sep 5 2013, http://www.r-bloggers.com/text-mining-the-complete-works-of-william-shakespeare/

[12]RdataMining.com: R and Data Mining, http://www.rdatamining.com/examples/text-mining

[13] steinbach M, Karypis, G. and kumar, V. "A comparison of Documents clustering Techniques", kdd workshop on text mining

[14] K.Naga Neerja, B.Prajna , "An effective Research Paper Recommender System based on Subspace Clustering", International Journal Of Engineering And Computer Science, Page No. 13306-13310 Volume 4, Issue 7, July 2015 .

[15] Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Data Set, Cambridge, 2011, DOI: http://dx.doi.org/10.1017/CBO9781139058452.002

[16] nptel.iitm.ac.in/courses/106104021/pdf_lecture/lecture30.pdf

[17] C. A. Murthy, Text Document Clustering, ACM Text Mining Workshop TMW-2014 at ISI Kolkata,

[18] Nadempalli Sneha, B.Prajna , Sharmila Sujatha , "Application for Retriving Details of Users - Topic Based Approach", pages 509-513 ,volume 6, Issue 8,IJCSET,2015

[19]https://rstudio-pubs-static.s3.amazonaws.com/31867_8236987cf0a8444e962ccd2aec46d9c3.html

[21] Han, J. and Kamber, M., *Data Mining: Concepts and Techniques, 2nd ed*., Morgan Kaufmann Publishers, 2006.

[22] Kaufman, L. and Rousseeuw, P., *Finding Groups in Data*, Wiley, New York, NY, 1990.

[23] *http://www.dit.ie/computing/research/resources/smsdata*