Journal of Theoretical and Applied Information Technology
10th February 2014. Vol. 60 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL APPROACH FOR TEXT CLUSTERING USING MUST LINK AND CANNOT LINK ALGORITHM

[1] J.DAFNI ROSE, [2] DIVYA D. DEV, [3] C.R.RENE ROBIN

[1] St.Joseph's Institute Of Technology, Department Of Computer Science And Engineering, Chennai-119
[2]St.Joseph's College Of Engineering, Department Of Computer Science And Engineering, Chennai-119
[3] Jerusalem College Of Engineering, Department Of Computer Science And Engineering, Chennai-100
E-mail: [1]jdafnirose@yahoo.co.in , [2]divyaddev@gmail.com , [3]crrenerobin@gmail.com

ABSTRACT

Text clustering is used to group documents with high levels of similarity. It has found applications in different areas of text mining and information retrieval. The digital data available nowadays has grown in huge volume and retrieving useful information from that is a big challenge. Text clustering has found an important application to organize the data and to extract useful information from the available corpus. In this paper, we have proposed a novel method for clustering the text documents. In the first phase features are selected using a genetic based method. In the next phase the extracted keywords are clustered using a hybrid algorithm. The clusters are classed under meaningful topics. The MLCL algorithm works in three phases. Firstly, the linked keywords of the genetic based extraction method are identified with a Must Link and Cannot Link algorithm (MLCL). Secondly, the MLCL algorithm forms the initial clusters. Finally, the clusters are optimized using Gaussian parameters. The proposed method is tested with datasets like Reuters-21578 and Brown Corpus. The experimental results prove that our proposed method has an improved performance than the fuzzy self-constructing feature clustering algorithm.

Keywords: *Genetic Algorithm, Keyword Extraction, Text Clustering, MLCL Algorithm.*

1. INTRODUCTION

Text mining is an important process in the field of information retrieval [1]. Text mining comprises of a wide range of processes like text clustering, classification, text summarization and automatic organization of text documents. Documents that are available on the internet are increasing day by day and most of them are loosely structured. Clustering has become a significant and widely used text mining tool to structure these documents so that similar documents are clustered into the same group and dissimilar documents are separated into different groups. [17]

Text clustering is an unsupervised learning method where similar documents are grouped into clusters. It is defined as method of finding groups of similar objects in the data. The similarity between objects is calculated using various similarity functions. Clustering can be very useful in various text domains, where the objects to be clustered are of various types such as paragraphs, sentences, documents or terms. Clustering helps to organize the documents which will further help to improve information retrieval and support browsing [4].

The quality of any text mining methods such as classification and clustering is highly dependent on the noisiness of the features that are used for the process. Therefore, the features should be selected effectively to improve the clustering quality. Some of the commonly used feature selection methods are document frequency based selection method, term strength and entropy based ranking. After the features have been selected, any text mining tasks such as classification, clustering, summarization can be applied [4].

The basic characteristics of text document include high dimensionality, sparsity and noisy features. The performance of the clustering algorithms is influenced by these properties. [17] Text clustering finds numerous applications in customer segmentation, classification, visualization, document organization and indexing.

The two main classifications of clustering algorithms are hierarchical based and K-means

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

based method. These are general purpose methods that can be extended to any kind of data [9]. The hierarchical method is divided into two types, namely, agglomerative and divisive [12]. The hierarchical method produces a hierarchical representation of the text documents. The method of agglomerative hierarchical clustering is particularly useful to support a variety of searching methods as it naturally creates a tree-like hierarchy, which can be leveraged for the search process. The main disadvantage of this method is its nature of irreversibility; i.e., once the text documents are merged or split, they cannot be rearranged due to their diminutive performance.

The process of text clustering usually contains two phases. The first phase is keyword extraction followed by clustering those keywords. For the first phase, the text documents are usually represented using the vector space model, where each row represents the documents, and the column corresponds to the various attributes of the document. As the document size increases, the performance of the VSM decreases [8]. To overcome this, in this paper, we propose a novel keyword extraction method and a clustering algorithm.

The basic requirements of a good clustering algorithm are:

1) The relationship between words should be displayed prominently in the document model.
2) Clusters must be identified with a meaningful label.
3) The high dimensionality of data has to be reduced efficiently.

Based on these requirements, we propose a genetic based keyword extraction method that reduces the document dimensionality. Keyword extraction is followed by the MLCL algorithm that will form meaningful clusters, which maintain the relationship between the key terms of a document [9].

In the proposed clustering algorithm, the relationship between words is identified using the Must Link and Cannot Link (MLCL) algorithm. Words which are similar to one another are grouped into the same cluster. Clusters are characterized by statistical deviation and mean values, using the membership function. The clusters which are formed are optimized using Gaussian parameters. The experimental results show that our method offers better performance.

The rest of the paper is organized as follows: Section 2 gives an overview of the research in text clustering. Section 3 discusses the proposed work of genetic based keyword extraction and the MLCL algorithm. Section 4 elaborates on the synopsis of the experimental results. Finally, section 5 concludes with a summary of the novel methods.

2. RELATED WORK

Feature clustering is an efficient approach for feature reduction, which groups all features into clusters, where features in a cluster are similar.The feature clustering methods proposed are “hard” clustering methods, where each word of the original features belongs to exactly one word cluster [6]. Therefore each word contributes to the synthesis of only one new feature. Each new feature is obtained by summing up the words belonging to one cluster. Let D be the matrix consisting of all the original documents with m features, from which D0, the matrix consisting of the converted documents, with new k features, can be formed.

Decision tree learners endeavour to select from training data some informative words using an information gain criterion, and then predict the category of a document based on the occurrence of word combinations. The most popular decision tree-based methods are based on word based techniques. Decision rule methods generate classifiers by inductive rule learning. While traditional machine learners employ attribute value representations, the use of logic programming representations led to the establishment of Inductive Logic Programming (ILP) [11]. The effectiveness of ILP methods for TC lies in formulating classifiers, based on the word order. One way to represent ordering information is with logic. Labelled training examples of the target class C can be represented as labelled ground facts of the form +c(d) or -c(d), where d is a constant that identifies a document, together with facts of the form Wi (d, p). The fact Wi (d, p) indicates that word Wi which appears in the document d at position p. In a typical ILP system, the c(d) facts will be used as training examples and the Wi (d, p) facts will be used as background relations.

Frequent Term-Based Clustering (FTC) is proposed for document clustering in [17]. The basic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

motivation of FTC is to produce document clusters with as few overlaps as possible. FTC works in a bottom-up fashion. Starting with an empty set, it continues selecting one more element (one cluster description) from the set of remaining frequent itemsets, until the entire document collection is contained in the cover of the set of all chosen frequent itemsets. In each step, FTC selects one of the remaining frequent itemsets, which has a cover with minimum overlap with the other cluster candidates, i.e., the cluster candidate which has the smallest entropy overlap (EO) value. The documents covered by the selected frequent item sets are removed from the collection D, and in the next iteration, the overlap for all the remaining cluster candidates is recomputed with respect to the reduced collection. The final clusters are then produced by the FTC method [17]. In FTC, a cluster candidate is represented by the frequent item sets and the documents in which they occur.

Clustering based on Frequent Word Sequence (CFWS) is proposed in [3]. The CFWS uses frequent word sequences and K-mismatch for document clustering. The difference between a word sequence and a word item set is that the word sequence considers the words' order, while the word item sets ignores the words' order. Suppose we have a document collection, with items in each document. Frequent sequences are extracted from these documents. There are overlaps in the final clusters of the CFWS [3].

The FIHC provides a tree for document clusters, which is easy to browse with meaningful cluster description [7]. Its characteristics of scalability and non-sensitivity to parameters are desirable properties for the clustering analysis. However, we conjecture that it has three disadvantages in practical applications. First, it cannot solve the cluster conflict when assigning documents to clusters. That is, a document may be partitioned into different clusters and this partition has a great influence on the final clusters produced by the FIHC [7]. Second, after a document has been assigned to a cluster, and the cluster frequent items are changed, the FICH does not consider these changes in a later overlapping measure. Third, in the FIHC, frequent item sets are used merely in constructing the initial clusters. Other processes in the FIHC, such as making clusters disjoint and pruning, are based on single items of documents, and decided by the initial clusters. One motivation of using frequent item sets is to use word co-occurrence of documents, and the co- occurrence of frequent item sets can furnish more information for clustering than the co-occurrence of single items.

K. Latha [10] proposes a heuristic approach called tabu annealing for the convergence of solution space for large sets of text documents and applies retrieval methodologies to find the information of interest. Tabu annealing is a combination of tabu search and simulated annealing with clustering approach. Jun-Peng Bao[9] proposes a heavy frequency vector which is developed as an improvement of the traditional VSM model. The heavy frequency vector considers only the most frequent words in a document. This method is found suitable for incremental clustering. In [14], a semantic enhanced hierarchical P2P is proposed to address the problem of modularity, flexibility and scalability in distributed P2P network.

3. PROPOSED ALGORITHM

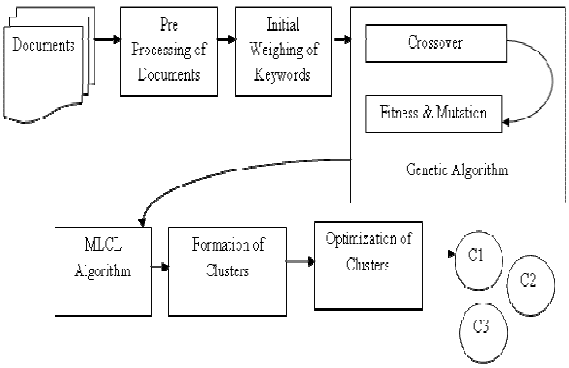

Fig 3.1 Overview of the Clustering Algorithm

Fig 3.1 shows the overview of the clustering algorithm. The proposed work is divided into two phases. The first phase is genetic based keyword extraction, which retrieves the important keywords from the documents. The keyword extraction algorithm displays how the genetic algorithm can be used to haul out the supreme set of keywords. This involves the repetitive application of the mutation, crossover, and fitness functions along with the selection operators. In the second phase the documents are clustered using Must Link and Cannot Link (MLCL) algorithm. The computational challenges have been analyzed for efficient and effective retrieval of the scoring words, from both the real and synthetic data sets.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1 Keyword Extraction

The initial phase of the keyword extraction involves pre processing of the document. The terms need to be assigned with a weight that will help in prioritizing them within the population. Once the initial weight is calibrated the genetic procedures are executed to gain a final keyword population. The terms are chromosomes and the weights are the numeric representation of genes. A simple modified arithmetic technique is applied for crossover, trialed by the "Expected Number of Elements in the Population" [10] viewpoint to declare the fitness of the engendered populace. Mutation is alleged only if the fitness utility is not contended for cessation.

The initial weighing equation is once again broken into two parts. When words occur for the first time, an extension of the weighed term standard deviation method, as given by [16] was used. The equation that was proposed previously exploits only certain document features. To improve the accuracy, the new formulation weighs the terms based on all the parameters, which could identify the terms in the document. The very first equation in the genetic based method is as follows. [5]

$$Vw_i = \frac{Wc_i / Wp_i}{D\max / Dc} \rightarrow (1)$$

Where
Vw_i – Weight of a word in position "I" of a sentence
Wc_i – Number of words in sentence "i"
Wp_i – average position of the word in sentence "i"
D_{max}– maximum number of words in a single line
D_c – The total number of words in a document.

When a word starts repeating, the next equation (2) is used. The equation has utilized the concept of term frequency. As a term keeps repeating over and over again, it could have a certain amount of relevance. The idea is made prominent by considering several other document attributes, which reveal the real relevance of each term. The formula at time "t+1" uses the weight that was being computed at time "t" and this is continued every time when a term reappears. The equations (2) and (3) now deviate from the base equations and navigate into the unique document features. This increases the accuracy of the output. [5]

$$W_{icr} = \frac{1}{Wc_i * \sqrt{Wp_i}} \rightarrow (2)$$

$$Vw_i = W_{icr} + \sqrt{Vw_i(n-1)} \rightarrow (3)$$

where
W_{icr}– value to be added to the weight of a word when it is encountered more than once
Vw_i (n-1) – Weight of the word at time n-1

This helps with the evaluation and words with lower weights give a reduced prominence to the document meaning. Words that occur deep down in the passage may be seen to have less weightage but the positional factors of the words in the particular sentence neutralize the drop. A balance is made between the position of the word in the document and its occurrences. The combination of the above two equations will produce the best set of keywords to be passed on to the next phase of the genetic algorithm.

3.1.1 Probability crossover

The basic principle behind crossover involves the divide and conquer method. The population is broken into two halves where the first segment contains the better half and the rest holds the weight of the lower probability population. W_i indicates the weight of a word in position "i". Pc_i is the probability ratio of the most feasible word with respect to the word that has the highest occurence in the other part of the division. The equations exploited in the crossover mechanism include the following: [5]

$$a_i = P_i / (K)$$

$$a_k = P_k / (K)$$

$$Pc_i = (a_i * P_i) + (1 - a_k) P_k \rightarrow (4)$$

$$Pc_k = (a_k * P_k) + (1 - a_i) P_i \rightarrow (5)$$

where
P_i- Weight of a word "i" in the first half of the population
P_k- Weight of a word "k" in the second half of the population
a_i- Probability occurrence of a word "i"
a_k- Probability occurrence of a word "k"
Pci- modified weight of the word "i"
Pc_k- modified weight of a word "k"

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The probability method of "p= (1-q)" is used to obtain the likeliness between words. "P" denotes the weight of a highly prioritized word and "Q" indicates words with a lower priority.

3.1.2 Fitness

The robustness of the solution depends on the selection of parents passed from the current process to the next iteration. The fitness function is used to generate a functional assessment of the comparative fitness expressions. The equation representation is as follows: [5]

$$E(n_i) = \frac{Pc(i)}{avg(Pc)} * D_{\max} \rightarrow (6)$$

$$T(i) = E(n_i) * D_{distinct\_terms\_in\_documents} \rightarrow (7)$$

$$if\ (T(i) < avg(F))$$

$$Pm_i = 0$$

where

$E(n_i)$ - Expected number of copies of "i"

Dc - Count of words in Document

Pc (i) - Crossover weight-age value of a word

Avg (Pc)-Average weight of all the words in the document

$D_{distinct\_terms\_in\_documents}$-Number of distinct words in the document.

3.1.3 Mutation

Mutation engrosses the amendment of term weights with a probability mutation "pm". Mutation has the capability to reinstate the mislaid genetic material into the population, thus thwarting the convergence of the solution into a suboptimal region or divergence into an infinite loop. T (i) decides whether the mutation process is to be applied or not. When a word does not lay within the fitness condition the process of mutation is being applied. The fitness value determines mutation. When two consecutive iterations have a similar weight-age for the terms, the ultimate keyword list is generated. If a word is fit, the mutation would not be applied. The equational representation of mutation is given by the following formula:

$$test\_val = Vw_i(i+1) - Vw_i(i)$$

$$Pm[i] = \sqrt{Vw_i(i)} + \frac{foundno[w]}{pos[w]} \rightarrow (8)$$

where

pos[w] - average of the overall position of the word w

foundno[w] - Word count of "w"

3.2 Clustering Process

In this phase, the prime attribute that is taken into consideration is the high dimensionality of the document space. The proposed system uses employs three different mechanisms. The first stage is the identification of related words in a document using the MLCL algorithm. The relevant keywords are grouped to form clusters using three main equations. Thereafter the clusters formed are optimized using Gaussian parameters. The Gaussian parameter identifies the word patterns and standard deviation of clusters. Then the words are grouped in accordance with the Gaussian outputs and colligated into clusters with reference to the documents.

3.2.1 Must Link and Cannot Link algorithm

The extracted keywords are passed into the MLCL algorithm. The terms which do not co-rrelate with the other terms are identified and eliminated using the MLCL algorithm. Each document is considered as an individual cluster of key terms. The equations used to calculate the similarity between the key terms of each document are based on the principles of cosine and Zipf's similarity.

Equations 9 and 10 compute a value for the "Must" link and the "Cannot" link, between one term of a document and the others. The values are compared against a threshold equation 11, to check if any kind of relationship can be established between the words. The related keywords will then be grouped to form a cluster. The must & cannot linkage has three different phases for the formation of clusters.

$$Dmin = W_1(t,d) - \sqrt{2 * W_2(t,d) * \log\left(\frac{1}{1 - \min(W(t,d))}\right)} \rightarrow 9$$

$$Dmax = \left(\log\left(\frac{1}{\max(W(t,d)) - 1}\right) * W_2(t,d)\right)^2 - W_2(t,d) \rightarrow 10$$

$$Val(Dmin) < Val(Dmax) \rightarrow 11$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The decision of whether the documents have to be grouped is done, with respect to its key terms. The key terms expressed in a numerical form will boost the rates of accuracy, but will not make the process of clustering easy. Like the other steps, this phase has a few "compelled" equations. The grouping of documents into clusters is based on equations 12 to 14. These equations are based on the weight of each term. The small variances can identify the apt words and place them in the accurate clusters. Clustering equations depend over the weight of each term. The three equations are as follows:

$$E1 = -\log \frac{|W1 - W2|}{Avg(W1, W2)} \quad (12)$$

$$E2 = \frac{Low(W1, W2)}{Avg(W1, W2)} \quad (13)$$

$$E3 = \frac{\log(low(W1, W2))}{\log(W1) + \log(W2)} \quad (14)$$

W1 and W2 are the terms in two different documents, D1 and D2, respectively. Equation 12 gives a numerical relationship between the terms w1 and w2 of two different documents. Equation 13 adopts the lowest weight amongst two words w1 and w2 to form a relationship between the two different documents. Equation 14 is a combined representation of E1 and E2. It gives the platform over which the relationship of the two terms can be calibrated. Based on these equations, the following inequalities are formed.

$$|E3 - E2| \leq |E1 - E2| \leq |E3 - E1|$$

Or

$$|E3 - E1| \leq |E1 - E2| \leq |E3 - E2|$$

Three different cases are considered based on these inequalities.

CASE 1: Documents with Matching Keywords
The two documents are grouped together
The average weight of the two key terms is computed in the final cluster

CASE 2: Clusters with matching the sub keywords
Clusters for matching sub keywords
Matched clusters will be grouped together
The average weight of the two key terms is computed in the final cluster

CASE 3: No Matching terms - Do Nothing

3.2.2 Algorithm for Optimization of Clusters

Let C be the total number of Clusters
Let words[p,o] be a word "p" in cluster "o"
Let N(j,k) = ϕ be the word "j" in cluster "k"
Let W(j) = ϕ be the word pattern of a term "j"
Let S(j) = ϕ be the standard deviation of a term "j"

```
Let Count(i) be the number of words in cluster "i"
for  i from 1 to C do
for j from i+1 to C do
for  x from 1 to count(i)  do
        for y from 1 to count(j) do
                if ( word[x,i] == word[ y,j])
                if( S(x) == S(y) )
                        Add S(y) to cluster "i"
                        Mark S(y) to 1
                End if
                End if
        End for
        End for
        End for
End for
Let flag=0
for  i from 1 to C do
        for  x from 1 to count(i)  do
                Get ( word[x,i] )
                        If (S(x) == 0) break
                        Else flag=1
        End For
                If flag =1 then delete cluster
End For
```

The algorithm 3.2.2 is used to optimize the clusters. It exhibits how the standard deviation and word pattern can be used to form clusters. Terms with similar standard deviation can be grouped into a single cluster and thus the output is optimized. Some documents can appear in more than one cluster. In such cases the clusters are optimized so that a document appears in one cluster alone.

4. TESTING

The performance of clusters is evaluated using Micro Measures like Micro Averaged Precision (MicroP), Micro Averaged Recall (MicroR), Micro Averaged F Measure (MicroF) and Micro Averaged Accuracy (MicroA).

Micro Averaged Precision [20, 21] is defined as the ratio of number of true positives (correct results) divided by the number of all returned results. It is the ability of the algorithm not to label as positive a sample that is negative.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$MicroP = \frac{\sum_{i=1}^{p} TP_i}{\sum_{i=1}^{p} (TP_i + FP_i)}$$

Micro Averaged Recall[20,21] is defined as the ratio of number of true positives correct results) divided by the number of results that shouldhave been returned. It is the ability of the algorithm to find all the positive samples.

$$MicroR = \frac{\sum_{i=1}^{p} TP_i}{\sum_{i=1}^{p} (TP_i + FN_i)}$$

Micro averaged F-measure [20, 21] is interpreted as the weighted average of precision and recall.

$$MicroF = \frac{2 * MicroP * MicroR}{MicroP + MicroR}$$

Micro Averaged Accuracy [22], is defined as the portion of all decisions that were correct decisions.

$$MicroA = \frac{\sum_{i=1}^{p} (TP_i + FN_i)}{\sum_{i=1}^{p} (TP_i + FP_i + TN_i + FN_i)}$$

Two different datasets the Reuters-21578, and the Brown Corpus were used for the study.

4.1 Reuters 21578

The Reuter-21578 has been widely used for testing clustering algorithms. Reuters-21578 is an experimental data collection that appeared on Reuters newswire of the year 1987. The dataset was obtained from http:kdd.ics.uci.edu/databases/reuters21578/reuters 21578.

Table 1 Micror Values

No of Documents	MLCL Clustering	Fuzzy Logic
25	94.32	77.76
35	97.95	63.35
45	97.77	61.23
50	98.65	59.79
60	99.047	57.14
75	99.2	55.98
100	99.08	61.18

Table 2 Microp Values

No of Documents	MLCL Clustering	Fuzzy Logic
25	82.34	63.45
35	88.79	58.02
45	93.29	45.505
50	93.95	45.22
60	94.28	54.29
75	90.40	57.49
100	92.09	55.71

Table 3 Microf Values

No of Documents	MLCL Clustering	Fuzzy Logic
25	87.92	69.87
35	93.69	60.56
45	95.477	52.20
50	96.24	51.49
60	96.60	55.67
75	94.59	56.72
100	95.45	58.31

Table 4 Microa Values

No of Documents	MLCL Clustering	Fuzzy Logic
25	77.66	74.33
35	87.94	36.755
45	91.20	78.22
50	82.68	70.37
60	83.33	31.02
75	89.67	32.18
100	91.24	34.08

The values of MicroR, MicroP and MicroF in tables 1 to 4 dictate on how the results of MLCL clustering stood ahead of Fuzzy logic, by numerical figures. This concludes the testing of single documents in a large data space with a common topic of identification.

4.2 Brown corpus

The next phase of testing is done on the other aspect of clustering. The real time application of clustering does “no” work on documents with individual topics. It will deal with documents of different topics, which will be combined together and tested for the effective formation of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

clusters. Here, each test comprises of five different document topics, and the proposed method works with the intention to form five distinguished clusters.

Brown Corpus has 500 sample English Documents. The document contains tagged words. It can be used to identify the tense of each word. The text sample is distributed over 15 different genres. Each sample starts with a random sentence boundary and continues to the next boundary.

Table 5 MicroR values

No of Documents	MLCL Clustering	Fuzzy Logic
25	95.43	74.23
35	91.34	87.88
45	88.65	90.23
50	90.32	56.62
60	87.68	78.45
75	93.22	50.02
100	93.03	50.02

Table 6 MicroP values

No of Documents	MLCL Clustering	Fuzzy Logic
25	30.33	22.22
35	54.67	45.89
45	62.55	60.71
50	70.97	53.01
60	74.55	75.64
75	56.67	45.99
100	65.05	72.46

Table 7 MicroF values

No of Documents	MLCL Clustering	Fuzzy Logic
25	46.03	34.20
35	68.40	60.29
45	73.34	72.58
50	79.48	54.75
60	80.58	77.01
75	70.48	47.92
100	76.56	59.18

Table 8 MicroA values

No of Documents	MLCL Clustering	Fuzzy Logic
25	28.94	16.49
35	49.92	40.32
45	55.45	54.77
50	64.10	30.01
60	65.36	59.33
75	52.82	23.00
100	60.51	36.24

The numerical values in tables 5 to 8 show that the new MLCL algorithm outperforms the fuzzy logic for brown corpus dataset also.

5. Conclusion

In this paper, clustering is done in two phases. First the dimensionality of the text document is decreased by selecting important keywords. Then the selected keywords are clustered using a MLCL algorithm. The novel method incorporates various computations to find the similarity between the words and the documents. The relationship between the words of a document is calculated using MLCL algorithm. Then similarity measures are used to identify the initial clusters and the clustering process is continued till all the documents are clustered. Finally the clusters are optimized using Gaussian parameters. The entire process is tested for its effectiveness with two different benchmark dataset. The new MLCL clustering algorithm is compared against Fuzzy self-constructing feature clustering and was found that the new novel method outperformed the existing algorithm in a consistent manner.

Our future research would be directed towards sentence based clustering which will be an extended version of our current work. The sentence based clustering will be used to improve the process of text summarization and clustering.

REFERENCES:

[1]. Andrew Skabar, KhaledAbdalgader, "Clustering Sentence Level Text using a Novel Fuzzy Relational Clustering Algorithm", *IEEE Transactions on Knowledge and Data Engineering*,2011, vol 25, issue 1, pp 62-75

[2]. Adriana Pietramala, Veronica L. Policicchio, Pasquale Rullo, InderbirSidhu," A Genetic Algorithm for Text Classification Rule Induction", *Springer Verlag Berlin Hiedelberypp*, 2008 188-203.

[3]. Beil.F, M.Ester, X.W.Xu," Frequent term based text clustering",. *Proceedings of the 8th ACM SIGKDD Int. conference on Knowledge discovery and data mining*. 2002,pp 436-442

[4]. Charu C.Aggarwal, ChengXiangZhai,"A Survey of Text Clustering Algorithms",Kluwer Academic Publishers.

[5]. Dafni Rose.J, Divya D.Dev, C.R. Rene Robin,"An improved Genetic based Keyword Extraction Technique",. *proceedings of Nature Inspired Cooperative Strategies for*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Optimization (NICSO 2013), Studies in Computational Intelligence 512 2013.DOI: 10.1007/978-3-319-01692-4_12.

[6]. Fei Liu, Feifan Liu, Yang Liu.A," Supervised Framework for keyword extraction from meeting transcripts", *IEEE Transactions on Audio, Speech and Language Processing*, 2011 Vol 19 pp 538-548

[7]. Fung, K.Wang, M. Ester," Hierarchial Document clustering using frequent itemsets", *proceedings of the 3rd SIAM Int. Conference on Data mining(SDM 2003).2003*

[8]. Hisham Al-Mubaid, Syed A. Umair,"A New Text Categorization and Technique using distributional clustering and learning Logic",*IEEE Transactions on Knowledge and Data Engineering*, 2006,Vol 18 Issue 9 pp 1156-1165

[9]. Jun-Peng Bao, Jun-Yi Shen, Xiao-Dong Liu, Hai-Yan Liu,"The heavy frequency vector-based text clustering", *International Journal of Business Intelligence and Data Mining*, 2005, Vol. 1, No.1 pp. 42 - 53.

[10]. Jung-Yi Jiang, Ren-JiaLiou, Shie-Jue Lee," A Fuzzy Self Constructing Feature Clustering Algorithm for Text Classification", *IEEE Transactions on Knowledge and Data Engineering*, 2011,Vol 23, Issue 3, pp 335-348

[11]. Latha.K, R. Rajaram,"Tabu annealing: an efficient and scalable strategy for document retrieval", *International Journal of Intelligent Information and Database Systems*,2009, Vol. 3, No.3 pp. 326 - 337

[12]. Meenakshi.M, K. Lakshmi, M. Saswati," A Negative category based approach for Wikipedia document", *Int J. Knowledge and Data Engineering*,2010,vol 1 pp 84-97.

[13]. Qirong Ho, Jacob Eisenstein, Eric P. Xing,"Document Hierarchies from Text and Links", *WWW2012 Entity and Taxonomy Extraction*,.2012, pp 739-748

[14]. Srinivas.M, L.M. Patnaik,"Adaptive Probabilities of Crossover and Mutation in Genetic Algorihms", *IEEE Transactions on Systems, man and Cybernetics*,1994,Vol 24, Issue 4, pp 656-667

[15]. Thangamani.M, P. Thangaraj,"Effective fuzzy semantic clustering scheme for decentralised network through multi-domain ontology model", *International Journal of Metadata, Semantics and Ontologies*, 2012 - Vol. 7, No.2 pp. 131 - 139

[16]. Wei Song, Cheng Hua Li, Soon Cheol Park,"Genetic algorithm for text clustering using ontology and evaluation the validity of various semantic similarity measures",*Expert Systems with applications*,2009 Vol 36, pp 9095-9104.

[17]. Wen Zhang, Taketoshi Yoshida, Xijin Tang, Qing Wang,"Text Clustering using frequent itemsets",*Knowledge Based Systems*,.2010Vol 23 pp 379-388

[18]. Weng S.S, Y.J Lin, .Jen,"A study on searching for similar documents based on multiple concepts and distribution of concepts",*Expert Systems with Applications* 2003,Vol 25, pp 355-368

[19]. Yunming Ye, Xutao Li, Biao Wu," A comparative study of feature weighting methods for document co-clustering",*Int. J. Information Technology, Communications and Convergence*, 2011, Vol. 1, No. 2.

[20]. [Online]http://scikitlearn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

[21]. [Online]http://en.wikipedia.org/wiki/F1_score

[22]. [Online]http://search.cpan.org/~kwilliams/Statistics-Contingency-0.08/Contingency.pm