Jurnal Teknologi

Full Paper

FIREFLYCLUST: AN AUTOMATED HIERARCHICAL TEXT CLUSTERING APPROACH

Athraa Jasim Mohammed[a,b*], Yuhanis Yusof[a], Husniza Husni[a]

[a]School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
[b]Information Technology Center, University of Technology, Baghdad, Iraq

Article history
Received
5 September 2015
Received in revised form
13 April 2017
Accepted
31 May 2017

*Corresponding author
s94734@student.uum.edu.my

Abstract

Text clustering is one of the text mining tasks that is employed in search engines. Discovering the optimal number of clusters for a dataset or repository is a challenging problem. Various clustering algorithms have been reported in the literature but most of them rely on a pre-defined value of the *k* clusters. In this study, a variant of Firefly algorithm, termed as FireflyClust, is proposed to automatically cluster text documents in a hierarchical manner. The proposed clustering method operates based on five phases: data pre-processing, clustering, item re-location, cluster selection and cluster refinement. Experiments are undertaken based on different selections of threshold value. Results on the TREC collection named TR11, TR12, TR23 and TR45, showed that the FireflyClust is a better approach than the Bisect K-means, hybrid Bisect K-means and Practical General Stochastic Clustering Method. Such a result would enlighten the directions in developing a better information retrieval engine for this dynamic and fast growing big data era.

Keywords: Firefly algorithm, clustering, data mining, swarm intelligence

Abstrak

Penggugusan teks merupakan salah satu tugas perlombongan teks yang digunakan dalam enjin carian. Penentuan bilangan gugusan yang optimum ialah satu permasalahan yang mencabar. Pelbagai algoritma penggugusan telah dilaporkan dalam kajian tetapi kebanyakan algoritma bergantung kepada nilai gugusan *k* yang perlu ditetapkan lebih awal. Dalam kajian ini, sebuah algoritma firefly, yang dinamakan FireflyClust telah dicadangkan untuk mengumpul dokumen teks secara automatik dalam bentuk hirarki. Pengoperasian algoritma penggugusan yang dicadangkan adalah berdasarkan kepada lima fasa: pra-pemprosesan data, penggugusan, melokasi semula item, pemilihan gugusan dan pembaikan gugusan. Eksperimen yang dilakukan adalah berdasarkan pelbagai nilai mula. Keputusan eksperimen ke atas koleksi TREC yang dikenali sebagai TR11, TR12, TR23 dan TR45, telah menunjukkan bahawa FireflyClust ialah kaedah yang lebih baik berbanding Bisect K-means, Bisect K-means hibrid dan kaedah penggugusan Practical General Stochastic. Keputusan seperti ini memberi petunjuk kepada pembangunan enjin capaian maklumat yang lebih baik untuk era *big data* yang dinamik dan pesat berkembang ini.

Kata kunci: Algoritma *firefly*, penggugusan, perlombongan data, kepintaran kerumunan

1.0 INTRODUCTION

Text clustering analysis is defined as classifying data objects into groups known as clusters, where each cluster includes similar objects that are different from another cluster [1, 2]. The aim of this analysis is to find clusters with high homogeneity (meaning high similarity between data objects in one cluster) and less heterogeneity (less similarity between clusters) [3]. There is a number of text clustering algorithms proposed over the years. These algorithms can be classified into two approaches based on the mechanism used to solve the problem; partitional and hierarchical [4, 5], and based on the initial information used in the clustering process (i.e. the number of clusters); static and dynamic [5].

Partitional clustering, such as K-means [6], clusters the data objects into a specific number of clusters that must be determined by a user as initial value. K-means is a powerful and simple method but it suffers

from local optima due the initial random centers, and it is not suitable when a prior knowledge on a dataset is absent.

Similar problem can also be seen in hierarchical clustering such as in the standard Bisect k-means [4, 7, 8, 9], where the number of clusters are used as the stopping criterion. Existing Bisect k-means operates by using K-means to group data objects in each level, and the Bisect k-means will later build the tree of clusters (in each level divided the current cluster into two children clusters). The working operation of Bisect k-means creates some issue as assigned data objects at a high level cluster cannot be moved to a lower level cluster. Nevertheless, it is reported in [9] that hierarchical clustering algorithm is better than partitional clustering in terms of cluster performance quality.

Recently, another approach in clustering has been introduced, which is swarm-based clustering. In swarm-based clustering, there are work focusing on solving the local optima problem in a static approach (where the number of k cluster is assumed) such as the one performed by Particle Swarm Optimization (PSO) [10], Ant Colony Optimization [11], Artificial Bee Colony [12], Cuckoo Optimization [13] and Wolf optimization [14]. Such an approach is efficient if knowledge about the datasets (i.e. the number of cluster) is known. Nevertheless, there are also working that focuses on solving the problem of automatically discovering the suitable number of clusters in dynamic approach. Where, Swarm based methods such as undertaken by Ant based clustering [15], Flocking based clustering [16] and Practical General Stochastic Clustering Method (PGSCM) [17] employ swarm like agents to cluster data directly without the requirement of defining the number of clusters. They adapt the mechanism of a specific insect or animal that exists in nature and convert it to heuristics rules [33].

Ant based clustering approach deals with the behavior of ants, where each ant performs sorting and corpse cleaning. This approach works by distributing the data object randomly in the 2D grid search space, then determining a specific number of ants (agents) that move randomly in this grid to pick up a data item if it does not hold any object (item) and drop the object (item) if it finds similar object. This process continues until it reaches a specific number of iterations [15]. PGSCM [17] is a simplification of ant based clustering approach. On the other hand, The Flocking based approach is related with behaviors of swarm intelligence [34] where a group of flocks swarm move in 2D or 3D search space following the same rules of flocks; get close to similar agents or far away from dissimilar agents [16]. This approach is computationally expensive as it requires multiple distance computations.

In the year of 2010, another nature-inspired algorithm termed as Firefly algorithm (FA) [18] was proposed by Xin-Shin Yang to solve optimization problem which later succeed in solving problems of diverse fields such as economic dispatch [19], allocation [20], image processing [21] and data clustering [3, 22, 37]. In this study, the standard Firefly algorithm is extended to automatically discover the optimal number of clusters and perform hierarchical clustering on datasets in the TREC collections [23].

1.1 Bisect K-means Clustering Algorithm

Hierarchical clustering approach involves two sub-approaches; agglomerative and divisive clustering algorithms [4, 9]. Agglomerative clustering approach operates by merging small clusters into a single cluster. This process builds a tree structure from bottom to top, where small clusters are available into the top. The Un-weighted Pair Group Method with Arithmetic mean (UPGMA) is one type of this approach and details on such work can be found in literature [8, 9]. On the contrast, Divisive clustering algorithm operates by splitting one big cluster into smaller clusters that builds a tree structure from top to bottom.

The Bisect K-means algorithm is an example of divisive hierarchical approach and it was presented by Steinbach *et al*. in 2000 [24]. The algorithm, at each level of hierarchy, classifies collection of objects into smaller groups and organizes clusters in a hierarchy. In [8, 9], at each level, Bisect K-means algorithm employs the K-means [6] to identify two clusters. This is followed by assigning objects in a dataset to the nearest center where the similarity is determined using Euclidean distance. The center of each cluster is calculated by identifying the mean. The process of assigning objects and recalculates the center continue until the stopping condition is reached. Figure 1 illustrates the process of K-means. At each level, the operation of choosing a cluster to split in Bisect K-means algorithm is based on some criterion such as minimum intra similarity [8, 9]. The process of Bisect K-means is illustrated in Figure 2. So, the drawbacks of the Bisect k-means are similar to K-means method; random initial centers which may cause trap in local optima, the predetermined the number of clusters [10].

Step 1: Randomly choose k cluster centers.
Step 2: Assign each object to closest center using Euclidean distance.
Step 3: Re-calculate the centers.
Step 4: Repeat Step 1 and Step 2 until stop condition is reached.

Figure 1 The process of K-means algorithm [6]

Step 1: Randomly choose two cluster centers.
Step 2: Clustering using K-means method, as shown in Figure 1.
Step 3: If does not reach number of clusters.
Step 4: Choose the cluster that has smallest intra similarity, Repeat Step 1 until reach number of clusters.

Figure 2 The process of Bisect K-means [8, 9]

In [27] has been applied a new method in K-means and Bisect K-means to identify the cluster centroids based on a new similarity measure that combine cosine function and link function (which is the number of common neighbors between two documents). The results were improved by adopting this method. Further, a cooperative approach between Bisect K-means and K-means has been presented in work [4], where, this approach combines the output results of Bisect K-means and K-means utilizing cooperative and merging matrices.

1.2 Hybrid Bisect K-means Clustering Algorithm

In this approach, a hybrid between divisive Bisect K-means algorithm (top-bottom tree) with agglomerative UPGMA algorithm (bottom-top tree) to address the problem of assigning documents to similar cluster in early stage and cannot be changed. This hybrid will correct the misplaced documents in the generated clusters [8, 9]. The hybrid approach operates initially by the whole dataset as one single cluster, then divides this single cluster into two sub-clusters using K-means algorithm. This process continues to work until generated number of clusters K' greater than original number K, then, computes the centroid for each cluster. This followed by calculating the similarity matrix (K' X K') between identified centroids. Merging two clusters in one cluster that have similar centroid, then, updating process is conducted by update the centroids and the similarity matrix. The merging step continues until generating the K original number of clusters. Figure 3 illustrates the pseudo code of this hybrid approach.
This approach generates good quality clusters compared to the standard Bisect K-means, however, it has problem with the number of clusters which is static and is predefined by the users.

1.3 Practical General Stochastic Clustering Method (PGSCM)

Practical General Stochastic Clustering Method (PGSCM) [17] is a dynamic clustering method that generates number of clusters without any prior information (i.e. the number of clusters). It is a simplification approach derived from nature-inspired ant-based clustering. Figure 4 illustrates the mentioned pseudo code.

Step 1: Choose a cluster to split.
Step2: Identify two sub-clusters using K-means algorithm.
Step 3: Repeat Steps 1 and 2 until generates k' number of clusters larger than k original number of clusters.
Step 4: Compute the centroids of identified clusters.
Step 5: Construct Similarity matrix between identified centers of clusters.
Step 6: Merging two clusters that have similarity between their centers.
Step 7: Update Steps 4 and 5.
Step 8: Repeat steps 6 and 7 until stop conditions is reached; the generating the k original number of clusters.

Figure 3 The pseudo code of hybrid Bisect K-means approach [9]

Step 1: Input the dataset D with n objects.
Step 2: The dissimilarity threshold is calculated for all n objects.
Step 3: Each object in dataset allocating to a bin.
Step 4: Do while iteration <= Max iteration
Step 5: Choose two objects from dataset D randomly and must not equal.
Step 6: If distance between two selecting objects < minimum dissimilarity threshold of two objects
Step 7: Store the comparison outcome.
Step 8: If the level of support (first object) < level of support (second object)
Step 9: Move first object to second object.
Step 10: Else Move second object to first object.
Step 11: End If
Step 12: End While
Step 13: output a set of clusters that represent all non-empty bins.

Figure 4 The pseudo code of PGSCM approach [17]

This approach succeeds to discover clusters in large datasets but in some real datasets that have large number of clusters with different size (non-normal distributed) such as Yeast, Zoo and Digits, it is discovered that they are far from optimal number of clusters. Hence, this study proposes a dynamic clustering based on Firefly algorithm that has the ability to discover near optimal clusters in non-normal distribution datasets.

1.4 Standard Firefly Algorithm

Firefly algorithm (FA) [18] is a swarm based algorithm that has the ability to identify global optimal solution efficiently. The idea of Firefly algorithm is based on two factors; light intensity and attractiveness between fireflies.

The light intensity of a firefly is related with the objective function f(x) and it can be a maximization or minimization problem. The attractiveness, β, between fireflies is related with light intensity and changes based on the distance between two fireflies. The process in the Firefly algorithm [18] is presented in Figure 5.

Step 1: Generate Initial population of firefly randomly xi (i=1, 2,.., n),Light Intensity I at xi is determine by Objective function f(xi).
Step 2: Define light absorption coefficient γ.
Step 3: While (t < Max Generation)
Step 4: For i=1 to N (N all fireflies)
Step 5: For j=1 to N
Step 6: If (Ii<Ij)
$\{ X^i = X^i + \beta_0 \exp^{(-Yr_{ij}{}^2)} * (X^j - X^i) + \alpha\varepsilon_i \}$
Step 7: $\beta = \beta_0 \exp^{(-Yr_{ij}{}^2)}$
Step 8: Evaluate new solutions and update light intensity.
Step 9: End For i
Step 10: End For j
Step 11: Rank the fireflies and find the current global best g*.
Step 12: End While

Figure 5 The pseudo code of standard Firefly Algorithm [18]

Firefly algorithm has been applied in many disciplines and proven to be successful in solving hard problems such as economic dispatch problem [29], image processing [21, 30], mobile network [31] and speech recognition [32]. Further, Firefly algorithm has been implemented effectively in numeric data clustering [3, 22].

2.0 METHODOLOGY

In this section, a proposed variant of Firefly algorithm, termed as FireflyClust, to be employed in text clustering is presented, where it includes additional phases (i.e item relocation and clusters refinement) and in phase (cluster selection) tested with two threshold. It initially starts with the dataset as a single cluster and without any prior knowledge about the dataset, and ended with the information on the optimal number of clusters. Furthermore, it also groups the documents into the identified clusters. The FireflyClust is performed in five phases: data pre-processing, clustering using Firefly algorithm, item re-location, clusters selection and clusters refinement. Figure 6 illustrates the framework of FireflyClust clustering method. Each phase is discussed in the following subsections.

2.1 Data Pre-processing

In this study, data pre-processing is the process of transforming a set of documents from unstructured into a structured form. The employed dataset undergoes four steps; data cleaning, stop words removal, word stemming and finally vector space model construction. In data cleaning, the selected texts from each document are extracted and cleaned from special characters and digits. After that, the cleaned texts undergo splitting processes that convert them into a set of words (set of terms). Further, the set of words (terms) are cleaned from words that have length less than three characters such as in, on, at, etc. In stop words removal, words such as propositions and conjunctions are removed. While in word stemming, all words (terms) are returned to the root such as the word 'playing' is returned to its root 'play' [26].

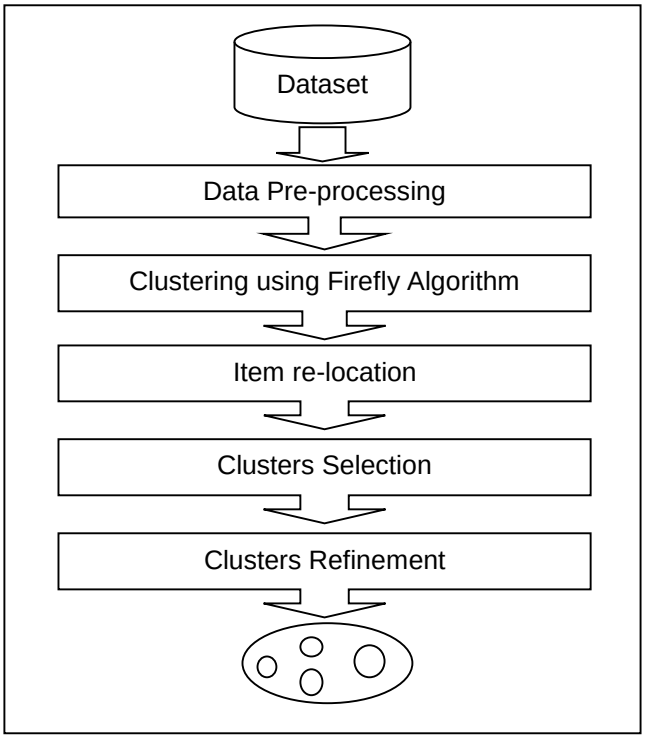

Figure 6 The framework of FireflyClust clustering method

Vector Space Model (VSM) commonly used in information retrieval and data mining approach where it represents the utilized data in a vector space [25]. In this phase, the obtained word are organized in a structure form in a Vector Space Model (VSM), where each document is represented as a vector D_n=[tf1, tf_2,..., tf_m] in the search space [25]. The vector space has two dimensions, *n* and *m*, where *n* denote the number of documents and *m* is the number of terms. The value of tf (term frequency) is represented by the number of term appeared in the document. Later, such information is transformed into term frequency-inverse document frequency (tf-idf). The benefit of using tf-idf is the balance between local and global term weighting in the document [26]. The value of tf-idf can be calculated using Eq. 1.

$$tfidf_{t,d} = tf_{t,d} * \log n/dft \quad (1)$$

Where, *n* represents the number of documents, dft refers to the number of documents that include a specific term.

2.2 Clustering using Firefly Algorithm

In this phase, as detailed in previous work [35, 36], each firefly represents a single document, where the light of a firefly initially indicates the weight of a document, and is obtained using Eq. 2.

$$totalwetight_{d_j} = \sum_{i=1}^{m} tfidf_{t_i,d_j} \quad (2)$$

The weight of a specific document is the sum of all tf-idf of the terms in that document. Each firefly has random real position in the search space and is in the range of (0, 1). The position is presented by coordinate (X, Y), and this study it is assumed that Y is static (initially is defined at 0.5) while the X is a random value between (0, 1). Later, distance between two documents (two real positions) is calculated using Cartesian distance function [18] as shown in Eq. 3.

$$CartesianDistance(X_i, X_j) = \sqrt[2]{(X_i - X_j)^2} \quad (3)$$

Further, we compute the similarity between documents using the Cosine function [27]. The value of cosine similarity is in the range of (0, 1), and when the value approaches 1, it indicates that the two documents are identical. On the other hand, if the value reaches 0, the two documents are far away and are not identical. Eq. 4 displays the formula to calculate the Cosine similarity. In this study, paper, Cosine similarity is based on the normalized term frequency value (term frequency normalize to length of documents).

$$CosineSimilarity(d_i, d_j) = \sum_{t=1}^{m} (d_{it} * d_{jt}) \quad (4)$$

Each firefly (document) competes with each other (documents) and the competition is based on two factors; brightness of the light (light intensity) and similarity. For the brightness, the firefly with a brighter light will attract the less bright firefly. As for similarity between fireflies, firefly with similarity value greater than a specific threshold (in experiment each dataset has different threshold) moves towards the brightest firefly as shown in the following pseudo code:

If Cosine Similarity (d_i, d_j) >= threshold

$$X_i new = X_i old + \beta(X_j - X_i old) + \alpha\varepsilon i$$

Where, β is the attractiveness between two documents and can be calculated by Eq. 5 [18]. During experiment, the initial value of β is set to 1 and the light absorption coefficient, Y, is set to 1. On the other hand, the εi is a random number calculated using Eqs. 6, 7 and 8.

$$\beta = \beta_0 exp^{(-Yr_{ij}{}^2)} \quad (5)$$

$$\varepsilon_i = random(MinTFIDF_{ij}, MaxTFIDF_{ij}) \quad (6)$$

$$MinTFIDF_{ij} = \alpha * (Min(TFIDF_i), Min(TFIDF_j)) \quad (7)$$

$$MaxTFIDF_{ij} = \alpha * (Max(TFIDF_i), Max(TFIDF_j)) \quad (8)$$

Where, *i* and *j* refer to documents, Min (TFIDF) refers to minimum document weight derived from TFIDF value, and Max (TFIDF) refers to maximum document weight derived from TFIDF value. This is followed by updating the light intensity of the brightest firefly (center) by Eq. 9 as shown in the following.

$$light\ intensity\ I\ (d_j) = I\ (d_j) + \beta \quad (9)$$

The competition between fireflies continues until it reaches a predefined number of iteration. Then, the process of sorting the firefly is performed where firefly with the brightness light is identified as the best point (represent the first center of a cluster). Once a center is determined, we assign documents that are similar (using the cosine similarity as illustrated in equation 4) to the chosen center and eliminates the document from the list. The process of finding a centroid and its cluster repeats for the remaining documents in the list of fireflies until all documents are grouped accordingly. The assignment process is based on a specific threshold (in the undertaken experiment, each dataset uses different threshold value).

2.3 Item Re-location

This phase starts to operate when the second cluster is constructed where the assignment of a document to a cluster relies on a pre-determined threshold. This means that documents that are close to the first identified centroid are assigned to the first cluster, however, these documents may be closer to any of the upcoming centroids. Such a situation will result poor purity. Hence, it is proposed that FireflyClust algorithm allows the re-location of an assigned document. The relocating of documents in the clusters, as illustrated in the pseudo code in Figure 7, operates when a new cluster, in second phase is constructed. This algorithm calculates the similarity (Cosine similarity) between the newly identified centroid (center of new cluster) and documents that have been assigned to other (previous) clusters. If the similarity value is higher, then the document is moved (re-locate) from the original cluster to the newly created cluster.

Step 1: Initial m=number of clusters.
Step 2: If m>=2
Step 3: For K=1 to (m-1)
Step 4: If length (current cluster (k))>1
Step 5: For z=1 to length (current cluster(k))
Step 6: If document (z) not equal center (k)
Step 7: If similarity (center (m),document(z)) greater than similarity (center(k), document (z))
Step 8: Move (z) from current cluster to recent cluster (m).
Step 9: end for
Step 10: end for

Figure 7 Pseudo code of item re-location Algorithm

2.4 Clusters Selection

The clusters produced from the two previous phases (clustering and re-location) have high purity but produces a large number of clusters. Hence, there is a need to identify pure clusters with large of number of documents and merge the non-selected clusters with the large ones. This process is achieved by choosing clusters that exceed an identified threshold (50, n/20). Where, n refers to the number of documents in a dataset. In practice, several attempts were made using the criterion of identified threshold as performed by Picarougne *et al*. (2007) and Tan *et al*. (2011) while the idea of merging clusters is adopted from Picarougne *et al*. (2007). The identified threshold (50, n/20) used by Tan *et al*. (2011) in PGSCM discover number of clusters far from optimal number of clusters. Hence, in this paper proposed another selected threshold (50, n/40) which is tested in experiments. The pseudo code of clusters selection is illustrated in Figure 8.

Step 1: Set selected threshold equal min (50, n/20), or min (50, n/40),
Step 2: For i= 1 to number of clusters
Step 3: If length (Ci) >= selected threshold
Step 4: Save Ci in selecting clusters.
Step 5: Else Save Ci in non-selecting clusters.
Step 6: End.

Figure 8 Pseudo code of clusters selection

2.5 Clusters Refinement

The output from the fourth phase is the clusters that exceed the pre-determined threshold. In this step, a new center (centroid) for each of the selected clusters is identified by calculating the sum of all tf-idf value of documents in the specific cluster and divided by number of documents in the cluster, as shown in Eq. 10. Later, documents contained in the non-selected cluster (small size clusters) are re-assigned to the nearest newly identified centroids using minimum distance between documents and center as shown in Eq. 11 and Eq. 12. The pseudo code for the relevant step is depicted in Figure 9.

Step 1: For i= 1 to k (number of selected clusters)
Step 2: Calculate the center for each cluster as shown in Eq.10.

$$Center(C_k) = \frac{\sum_{j=1}^{NC_k} TFIDF_{D_j}}{NC_k} \quad (10)$$

Step 3: End For i
Step 4: For i= 1 to (number of documents in non-selected clusters)
Step 5: Find minimum distance between document Di and center of C1 using Eq. 11 as shown below.

$$mindistance(D_i, Center_{C_1}) = \sum_{j=1}^{m} (D_{ij} - Center_{C_1})^2 \quad (11)$$

Step 6: Assign Di=1
Step 7: For k= 2 to (number of selected clusters)
Step 8: Find minimum distance between document Di and center of Ck using Eq. 12 as shown below.

$$\text{mindistance2}(\text{D}_\text{i}, \text{Center}_{\text{C}_\text{k}}) = \sum_{\text{j=1}}^{\text{m}} (\text{D}_\text{ij} - \text{Center}_{\text{C}_\text{k}})^2 \quad (12)$$

Step 9: If (mindistance >= mindistance2), Assign Di=k, mindistance = mindistance2
Step 10: End For k
Step 11: Assign Di to Ck
Step 12: End For i

Figure 9 Pseudo code of clusters refinement

3.0 RESULTS AND DISCUSSION

3.1 Experiments Setup

In order to evaluate the proposed clustering algorithm, results on the performance of four methods; proposed FireflyClust, standard Bisect K-means [8, 9], Hybrid Bisect K-means [8, 9] and Practical General Stochastic Clustering Method (PGSCM) [17] is compared. The comparison is made based on external clustering indices and statistical analysis of Independent samples t-test. Experiments of FireflyClust and PGSCM were carried out in Matlab on windows 8 with a 2000 MHz processor and 4 GB memory. The algorithms were executed for ten times with twenty numbers of iterations, and the result is based on the mean values. On the other hand, the result of standard Bisect K-means and Hybrid Bisect K-means is obtained from [8, 9].

3.1.1 Evaluation Metrics

The employed external indices include the Purity, F-measure and Entropy. The Purity clustering quality is the measure of the extent of the cluster that includes only one class of data objects [8, 9]. Further, it defines as the maximal precision value for every class. The

higher the value of purity, the higher the clustering quality is. Eq.13 and Eq.14 is used to compute the purity which is based on the maximum number of documents that carry the class Θ_k in the cluster C_j respectively.

$$Purity = \sum_{\Theta_k \in \{\Theta_1,\dots,\Theta_c\}} \frac{P(\Theta_k, C_j)}{N} \quad (13)$$

$$P(\Theta_k, C_j) = Max_k \ |\Theta_k \cap C_j| \quad (14)$$

On the other hand, the F-measure is the measuring of the test's accuracy. It is based on two important metrics that is mostly used in Information Retrieval which are; Precision and Recall [8, 9, 25]. Precision is the number of members of the class Θ_k in the cluster C_j divided by the number of members of cluster C_j as shown in Eq.15, while, Recall is the number of the members of the class Θ_k in the cluster C_j divided by the actual number of members of class Θ_k in the dataset as shown in Eq.16. The total F-measure is the sum of maximum accuracy (F-measure) of individual class weighted according to the class size. It is shown as in Eq. 17 and Eq. 18.

$$Precision(\Theta_k, C_j) = \frac{|\Theta_k \cap C_j|}{|C_j|} \quad (15)$$

$$Recall(\Theta_k, C_j) = \frac{|\Theta_k \cap C_j|}{|\Theta_k|} \quad (16)$$

$$F(\Theta_k) = \max_{C_j \in \{C_1,\dots,C_k\}} \left(\frac{2 * R(\Theta_k, C_j) * P(\Theta_k, C_j)}{R(\Theta_k, C_j) + P(\Theta_k, C_j)} \right) \quad (17)$$

$$\text{Total Fmeasure} = \sum_{k=1}^{C} \frac{|\Theta_k|}{N} * \max(F(\Theta_k)) \quad (18)$$

The Entropy measures the goodness of clusters and randomness [8, 9, 28]. Additionally, Entropy can be defined as the measurement of the classes' distribution in each cluster. When the clustering solution involves a single class of documents in each cluster, it leads to less distribution of classes in cluster and low entropy value. Such a result indicates high quality performance of clustering. Eq. 19 provides the entropy of output cluster C_j which is the sum of probability distribution of classes in cluster C_j, while Eq.20 defines the total entropy for a clustering algorithm which equals the sum of single cluster entropies weighted according to the cluster size.

$$HC_j = -\sum_{k=1}^{C} \frac{|\Theta_k \cap C_j|}{|C_j|} log \frac{|\Theta_k \cap C_j|}{|C_j|} \quad (19)$$

$$H = \sum_{j=1}^{k} \frac{HC_j * |C_j|}{N} \quad (20)$$

Finally, this study also undertakes statistical analysis of Independent samples t-test on the mean difference between the pairs of FireflyClust and PGSCM using all metrics.

3.1.2 Document Data Sets

This study is realized on four datasets retrieved from TREC-5, TREC-6 and TREC-7 collections. These datasets named TR11, TR12, TR23 and TR45 [23]. All data sets are available at http://trec.nist.gov/ and Table 1 summarizes the characteristics of these datasets.

Table 1 Description of TREC Collection Data

Dataset	No. of Documents	No. of Classes	Min no. of Documents in Class	Max no. of Documents in Class	No. of Terms
TR11	414	9	6	132	6429
TR12	313	8	9	93	5804
TR23	204	6	6	91	5832
TR45	690	10	14	160	8261

3.2 Experimental Results

This section includes two experimental results; first is the comparison of evaluation metrics obtained by the proposed FireflyClust using different selection thresholds. We named the FireflyClust adopting selection (50, n/20), as in [16, 17], as FireflyClust1 and FireflyClust2 is the algorithm utilizing (50, n/40) selection threshold.

Second, is the comparison of evaluation metrics of the proposed FireflyClust methods and state of art methods; standard Bisect K-means [8, 9], Hybrid Bisect K-means [8, 9] and Practical General Stochastic Clustering Method (PGSCM) [17].

3.2.1 Results and Discussion of FireflyClust1 and FireflyClust2

Table 2 tabularizes the obtained results of Purity, F-measure and Entropy for FireflyClust1 and FireflyClust2. From the table, it is noted that the FireflyClust2 has higher value of Purity (0.6051, 0.4947, 0.5588, and 0.5596) in all datasets compared to FireflyClust1. Further, the FireflyClust2 has higher F-measure (0.4705, 0.4058 and 0.4769) which it is generated using TR11, TR12 and TR45, while FireflyClust1 has higher F-measure (0.4127) only in TR23. FireflyClust2 generates the best result in Entropy (i.e 1.6761, 1.8550, 1.5605 and 1.9449) for all datasets. Based on literature [8, 9], it is learned that a good clustering solution is the one with F-measure and purity values approaching to 1 and Entropy value approaching to 0.

The number of generated clusters in FireflyClust2 is near actual clusters, that means the selection threshold (50, n/40) is the best to be used with this

dataset (the classes of the dataset is not normally distributed).

Table 2 Results: FireflyClust1 vs. FireflyClust2, bold value is best

Data Sets and actual number of clusters	FireflyClust$_1$				FireflyClust$_2$			
	Purity	F-measure	Entropy	# Clusters	Purity	F-measure	Entropy	# Clusters
TR11 (9)	0.5401	0.4565	1.8671	5	**0.6051**	**0.4705**	**1.6761**	**8.6**
TR12 (8)	0.4319	0.3657	2.0622	5.8	**0.4946**	**0.4058**	**1.8550**	**8.1**
TR23 (6)	0.5554	0.4127	1.6227	5	**0.5588**	**0.4108**	**1.5605**	**6**
TR45 (10)	0.4416	0.4213	2.4305	4	**0.5596**	**0.4769**	**1.9449**	**9**

Hence, the FireflyClust2 algorithm produces a better quality performance; F-measure, Purity and Entropy and also the optimal number of clusters. This suggests that FireflyClust2 algorithm is a better algorithm and a more compact clustering as compared to FireflyClust1 method.

Figure 10 displays the graphical results of the external indices; Purity between FireflyClust1 and FireflyClust2 algorithms using different datasets (TR11, TR12, TR23 and TR45).

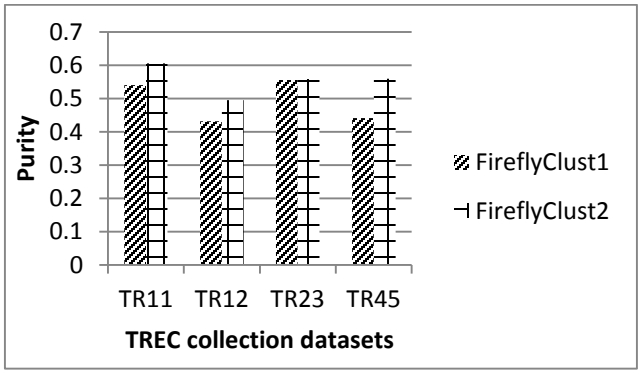

Figure 10 Graphical results of Purity metric between FireflyClust1 vs. FireflyClust2 using TREC collection datasets

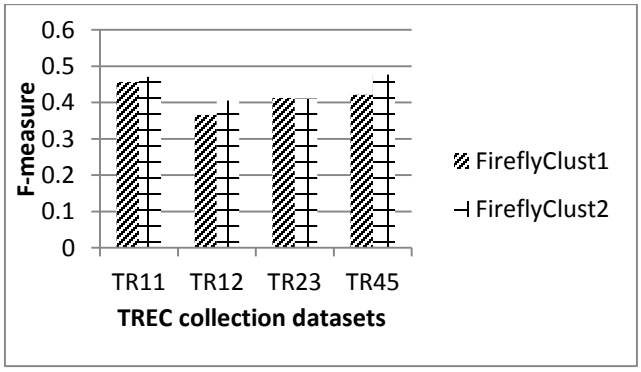

Figure 11 Graphical results of F-measure metric between FireflyClust1 vs. FireflyClust2 using TREC collection datasets

Figure 11 shows the graphical results of the external indices; F-measure between FireflyClust1 and FireflyClust2 algorithms using different datasets (TR11, TR12, TR23 and TR45).

Figure 12 shows the graphical results of the external indices; Entropy between FireflyClust1 and FireflyClust2 algorithms using different datasets (TR11, TR12, TR23 and TR45).

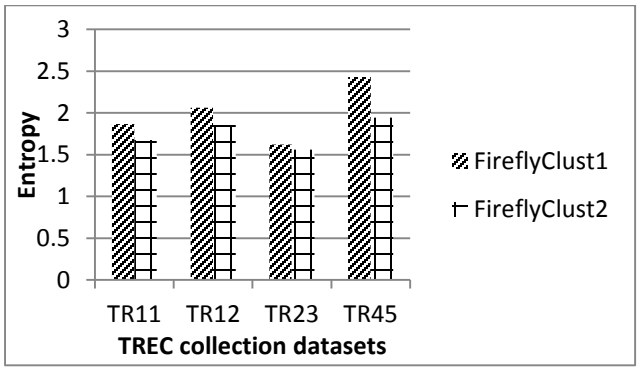

Figure 12 Graphical results of Entropy metric between FireflyClust1 vs. FireflyClust2 using TREC collection datasets

The number of generated clusters in FireflyClust1 and FireflyClust2 algorithms using different datasets (TR11, TR12, TR23 and TR45) are illustrated in graphical results in Figure 13.

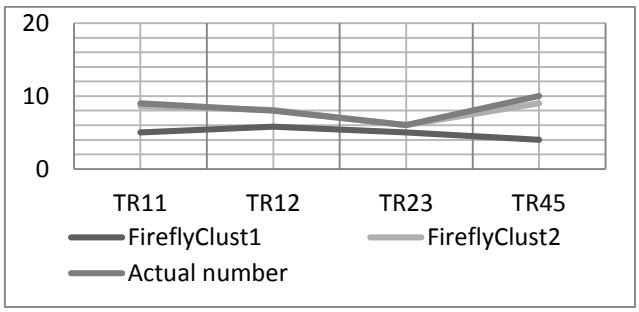

Figure 13 Graphical results of the number of generated clusters between FireflyClust1 vs. FireflyClust2 using TREC collection datasets

3.2.2 Results and Discussion of FireflyClust with Others Methods

Table 3 tabularizes the experimental results of Purity, F-measure and Entropy for four algorithms, the proposed FireflyClust2, Bisect K-means, hybrid Bisect K-means and PGSCM. As can be seen from Table 3, the Purity value for FireflyClust2 is higher than the other methods in all datasets. It is noted that in TR11 dataset, the highest purity is 0.6051 and was obtained by FireflyClust2, while the static methods, Bisect K-means and hybrid Bisect K-means generated 0.4850 and 0.4894 and the dynamic method PGSCM achieved a smaller value of 0.3327. In TR12 dataset, the proposed FireflyClust2 produces 0.4947, while Bisect K-means generates 0.3514, hybrid Bisect K-mean with 0.3837 and PGSCM settle at 0.3022. In TR23, the purity value is (0.5588, 0.4853, 0.5113 and 0.4475) for FireflyClust2, Bisect K-means, hybrid Bisect K-means and PGSCM respectively.

Table 3 Results: FireflyClust2 vs. Bisect K-means vs. hybrid Bisect K-means vs. PGSCM, bold value is best

Validity Indices	Datasets	FireflyClust2	Bisect K-means	Hybrid Bisect K-means	PGSCM
Purity	TR11	**0.6051**	0.4850	0.4894	0.3324
	TR12	**0.4946**	0.3514	0.3837	0.3022
	TR23	**0.5588**	0.4853	0.5113	0.4475
	TR45	**0.5596**	0.4210	0.4774	0.2652
F-measure	TR11	**0.4705**	0.2478	0.2910	0.2566
	TR12	**0.4058**	0.1946	0.2928	0.2334
	TR23	**0.4108**	0.1719	0.3217	0.3341
	TR45	**0.4769**	0.2627	0.3981	0.2478
Entropy	TR11	1.6761	1.4102	**1.4011**	2.5693
	TR12	1.8550	1.7344	**1.3798**	2.6668
	TR23	1.5605	1.3351	**1.2071**	2.0447
	TR45	1.9449	1.5922	**1.4059**	2.9131
# clusters	TR11	**8.6**	9	9	6.8
	TR12	**8.1**	8	8	5.9
	TR23	**6**	6	6	4.1
	TR45	**9**	10	10	4.4

In TR45, FireflyClust2 generates 0.5596 and it is better than Bisect K-means that only present users with is 0.4210, the hybrid Bisect K-means is at 0.4774 and PGSCM produces 0.2652. Figure 14 displays the graphical results of the external indices; Purity between FireflyClust2 and other algorithms using different datasets (TR11, TR12, TR23 and TR45).

Further, it is noted from Table 3 that the F-measure of FireflyClust2 outperformed the Bisect K-means, hybrid Bisect K-means and PGSCM in all datasets where the best F-measure value is (0.4058, 0.4705, 0.4108 and 0.4769) generated by FireflyClust2 in TR11, TR12, TR23 and TR45 respectively.

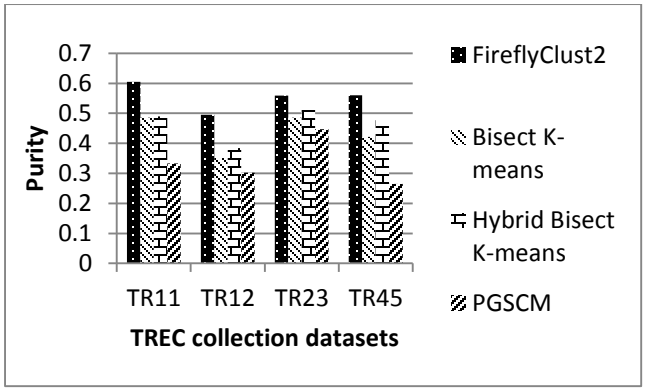

Figure 14 Graphical results of Purity metric between FireflyClust2 vs. Bisect K-means vs. hybrid Bisect K-means vs. PGSCM using TREC collection datasets

On the other hand, the F-measure value for PGSCM is better than Bisect K-means (0.2566, 0.2334, 0.3341 and 0.2478) in TR11, TR12, TR23 and TR45 respectively. However, it is learned that the F-measure value of hybrid Bisect K-means is better than Bisect K-means in all datasets and is better than PGSCM in most datasets (refer to TR11, TR12 and TR45 datasets). As a higher value of F-measure indicates a better algorithm, it can be concluded that FireflyClust2 is a better algorithm as compared to its competitors. Figure 15 shows the graphical results of the external indices; F-measure between FireflyClust2 and other algorithms using different datasets (TR11, TR12, TR23 and TR45).

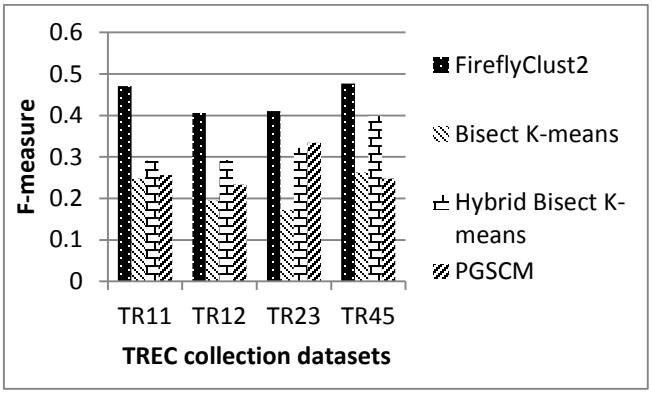

Figure 15 Graphical results of F-measure metric between FireflyClust2 vs. Bisect K-means vs. hybrid Bisect K-means vs. PGSCM using TREC collection datasets

In addition, the FireflyClust2 has best Entropy against dynamic method; PGSCM in all datasets, where the best value of FireflyClust2 is (1.6761, 1.8550, 1.5605 and 1.9449) produced in TR11, TR12, TR23 and TR45 respectively. Further, we can observe that hybrid Bisect K-means method is better than all methods including our proposed FireflyClust2 in generating lower Entropy (1.4011, 1.3798, 1.2071 and 1.4059). Figure 16 shows the graphical results of the Entropy indices among FireflyClust2, Bisect K-means, hybrid Bisect K-means and PGSCM.

As can see in Table 3, the number of produced clusters by dynamic FireflyClust2 is (8.6, 8.1, 6 and 9) which is near to the optimal clusters in datasets (TR11, TR12, TR23 and TR45) respectively, and is better than dynamic PGSCM that produces smaller numbers than actual cluster number. Figure 17 shows the graphical representation of the number of clusters in FireflyClust2, Bisect K-means and. PGSCM using TREC collection datasets.

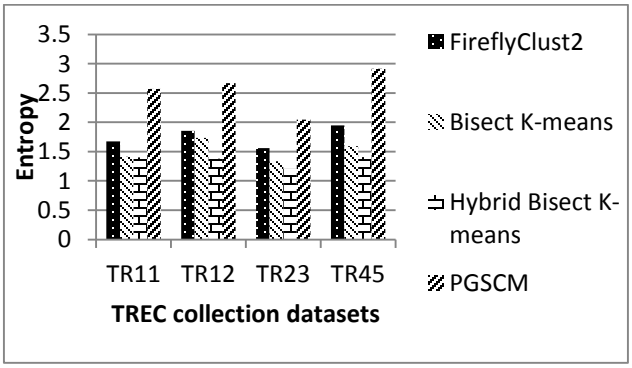

Figure 16 Graphical results of Entropy metric between FireflyClust2 vs. Bisect K-means vs. hybrid Bisect K-means vs. PGSCM using TREC collection datasets

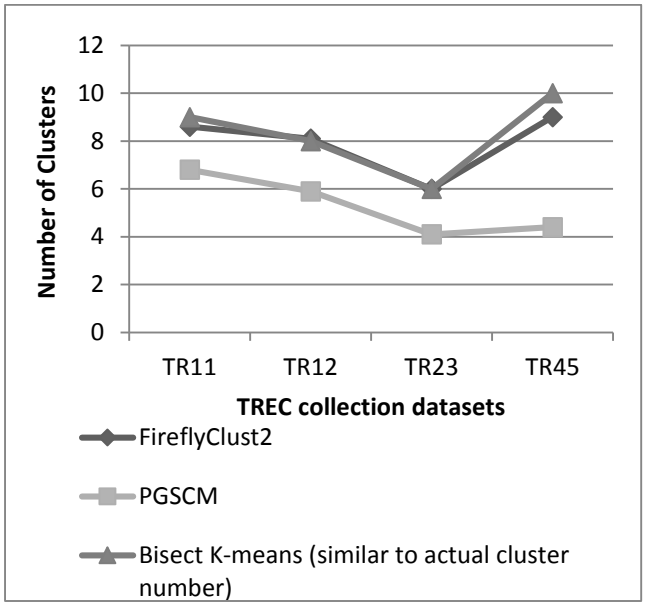

Figure 17 Graphical results of the number of generated clusters between FireflyClust2 vs. Bisect K-means vs. PGSCM using TREC collection datasets

3.2.3 Statistical Result

The statistical analysis of Independent Samples T-test is performed on the differences between the mean of two algorithms. In this study, the undertaken tests were between FireflyClust2 and PGSCM using all metrics. It is assumed that the null hypotheses and the alternative hypotheses are as shown below:

H_0: There is no difference between the mean of two algorithms.
H_1: There is a difference between the mean of two algorithms.

Table 4 reports the p-value using the samples of purity, F-measure and Entropy metrics between FireflyClust2 and PGSCM. The p-value is used to determine statistically the significance of the results.

Table 4 The p-value between FireflyClust2 & PGSCM

Datasets	Purity	
	Equal variances assumed	Equal variances not assumed
TR11	1.2463E-13	1.3707E-09
TR12	8.6277E-24	1.5073E-20
TR23	1.1974E-26	2.9152E-15
TR45	4.2291E-15	6.4004E-10
Datasets	**F-measure**	
	Equal variances assumed	**Equal variances not assumed**
TR11	1.5749E-09	2.6364E-08
TR12	1.4329E-12	1.0149E-10
TR23	2.2567E-07	2.1135E-05
TR45	5.8008E-13	6.1479E-10

Datasets	Entropy	
	Equal variances assumed	Equal variances not assumed
TR11	2.9890E-14	2.6680E-10
TR12	4.1752E-22	1.0781E-21
TR23	5.3008E-25	1.9432E-14
TR45	2.7661E-16	1.3980E-10

As can be observed in the table, i.e. Table 4, the p-value between FireflyClust2 and PGSCM is less than the cutoff value (0.05). This indicates strong evidence against the null hypothesis, so the null hypothesis that there is no difference between the mean of FireflyClust2 and PGSCM algorithms is rejected, hence suggesting that there is a difference between the mean of FireflyClust2 and PGSCM.

3.2.4 Scalability of FireflyClust2

Scalability refers to the ability of the system or method to continue operating efficient as it is varied in size or volume. It would have a linear growth with the size of input. In this study, the scalability of proposed FireflyClust2 is tested with performance metrics by changing the dimension of datasets. Figures 18, 19 and 20 show a graphical representation of scalability of proposed FireflyClust2 using purity, F-measure and entropy respectively.

As can be seen in Figure 18 the curve of purity is stable when changing the dimension of TREC collection datasets, while in Figure 19 can see the curve of F-measure increase when changing the dimension, and also in Figure 20 can observe the stable of the entropy curve between 1.5 and 1.9. From previous results conclude that proposed FireflyClust2 has good scalability as the number of dimensions in the data increase.

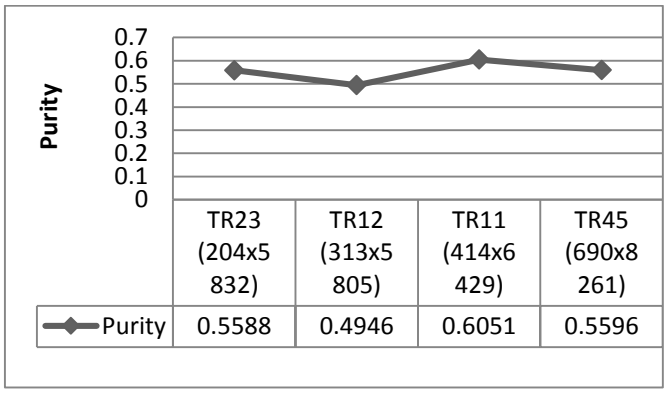

Figure 18 Graphical results of scalability of proposed FireflyClust2 using purity

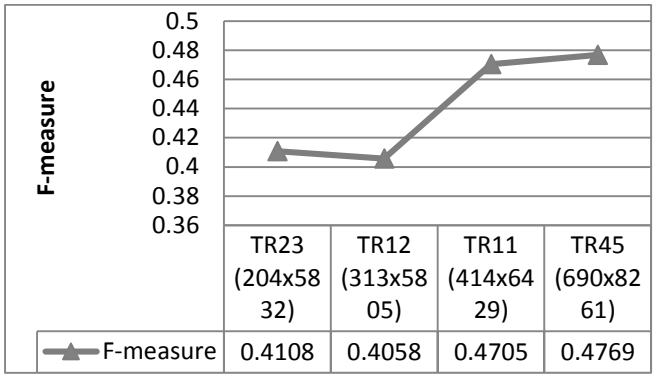

Figure 19 Graphical results of scalability of proposed FireflyClust2 using F-measure

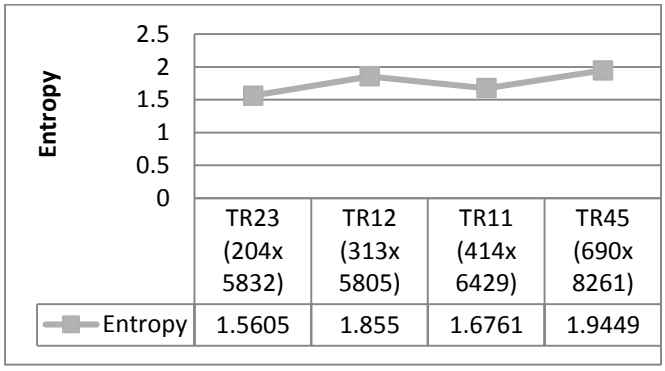

Figure 20 Graphical results of scalability of proposed FireflyClust2 using Entropy

4.0 CONCLUSION

In this paper, a new FA variant for hierarchical text clustering, named FireflyClust, is proposed. The novelty of this study is that the FireflyClust offers the re-locating procedure that is aimed to improve clustering purity. Further, it produces optimal number of clusters by invoking the selecting and refining phases. Strong and large clusters are merged with the smaller ones in order to reduce the number of clusters. Results obtained indicated that the proposed FireflyClust is a better approach as compared to existing Bisect K-means, hybrid Bisect K-means and PGSCM. It works well with non-normal distributed data and this is useful in the area of information retrieval. As users are now presented with overloaded information, automatically grouping unknown and dynamic datasets or repositories would facilitate searching and retrieval process.

Acknowledgement

Authors would like to thank the Ministry of Higher Education for providing the financial support under the Fundamental Research Grant Scheme (s/o: 12894).

References

[1] Hu, G., Zhou, S., Guan, J., and Hu, X. 2008. Towards Effective Document Clustering: A constrained K-means based Approach. *Elsevier, Information Processing & Management*. 44(4): 1397-1409.

[2] Aliguliyev, R. M. 2009. Performance Evaluation of Density-based Clustering Methods. *Elsevier, Information Sciences*. 179(20): 3583-3602.

[3] Banati, H., and Bajaj, M. 2013. Performance Analysis of Firefly Algorithm for Data Clustering. *Int. J. Swarm Intelligence*. 1(1): 19-35.

[4] Kashef, R., and Kamel, M. 2010. Cooperative Clustering. *Elsevier, Pattern Recognition*. 43(6): 2315-2329.

[5] Gil-Garicia, R., and Pons-Porrata, A. 2010. Dynamic Hierarchical Algorithms for Document Clustering. *Elsevier, Pattern Recognition Letters*. 31(6): 469-477.

[6] Jain, A. K. 2010. Data Clustering: 50 Years Beyond K-means. *Elsevier, Pattern Recognition Letters*. 31(8): 651-666.

[7] Kashef, R., and Kamel, M. S. 2009. Enhanced Bisecting k-means Clustering using Intermediate Cooperation. *Elsevier, Pattern Recognition*. 42(11): 2557-2569.

[8] Murugesan, K., and Zhang, J. 2011. Hybrid Bisect K-means Clustering Algorithm. *International Conference on Business Computing and Global Informatization*. 29-31.

[9] Murugesan, K., and Zhang, J. 2011. Hybrid Hierarchical Clustering: An Experimental Analysis (No. CMIDA-HIPSCCS#001-11). University of Kentucky.

[10] Cui, X., Potok, T. E., and Palathingal, P. 2005. Document Clustering using Particle Swarm Optimization. *IEEE Swarm Intelligence Symposium, SIS 2005*. 185-191.

[11] He, Y., Hui, S. C., and Sim, Y. 2006. Anovel Ant-based Clustering Approach Document Clustering. In H. Tou Ng, M. K. Leong, M. Y. Kan and D. Ji (Eds.). Information Retrieval Technology Springer Berlin Heidelberg. 4182: 537-544.

[12] Karaboga, D., and Ozturk, C. 2011. A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm. *Elsevier, Applied Soft Computing*. 11(1): 652-657.

[13] Zaw, M. M., and Mon, E. E. 2013. Web Document Clustering Using Cuckoo Search Clustering Algorithm based on Levy Flight. *International Journal of Innovation and Applied Studies*. 4(1): 182-188.

[14] Rui, T., Fong, S., Yang, X. S., and Deb, S. 2012. Nature-Inspired Clustering Algorithms for Web Intelligence Data. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. 147-153.

[15] Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chrétien, L. 1991. The Dynamics of Collective Sorting: Robot-like Ants and Ant-like. *The First International Conference On Simulation Of Adaptive Behavior On From Animals To Animates*.

[16] Picarougne, F., Azzag, H., Venturini, G., and Guinot, C. 2007. *A New Approach of Data Clustering Using a Flock of Agents*. Evolutionary Computation, Cambridge: MIT Press. 15(3): 345-367.

[17] Tan, S. C., Ting, K. M., and Teng, S. W. 2011. A General Stochastic Clustering Method for Automatic Cluster Discovery. *Elsevier, Pattern Recognition*. 44(10-11): 2786-2799.

[18] Yang, X. S. 2010. *Nature-Inspired Metaheuristic Algorithms*. 2nd Edition. United Kingdom: Luniver Press.

[19] Yang, X. S., Hosseini, S. S. S., and Gandomi, A. H. 2012. Firefly Algorithm for Solving Non-Convex Economic Dispatch Problems with Valve Loading Effect. *Elsevier, Applied Soft Computing*. 12(3): 1180-1186.

[20] Dos Santos Coelho, L., de Andrade Bernert, D. L., and Mariani, V. C. 2011. A Chaotic Firefly Algorithm Applied to Reliability-Redundancy Optimization. *IEEE Congress on Evolutionary Computation (CEC), New Orleans*. 517-521.

[21] Horng, M. H., and Jiang, T. W. 2010. Multilevel Image Thresholding Selection based on the Firefly Algorithm. *The 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), Xian, Shaanxi*. 58-63.

[22] Senthilnath, J., Omkar, S. N., and Mani, V. 2011. Clustering using Firefly Algorithm: Performance Study. *Elsevier, Swarm and Evolutionary Computation*. 1(3): 164-171.

[23] TREC. 1999. Text Retrieval Conference (TREC).

[24] Steinbach, M., Karypis, G., and Kumar, V. 2000. A Comparison of Document Clustering Techniques. The KDD workshop on Text Mining, Boston.

[25] Aliguliyev, R. M. 2009. Clustering of Document Collection-A Weighted Approach. *Elsevier, Expert Systems with Applications*. 36(4): 7904-7916.

[26] Manning, C. D., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. 1 ed. Cambridge University Press.

[27] Luo, C., Li, Y., and Chung, S. M. 2009. Text Document Clustering Based on Neighbors. *Elsevier, Data & Knowledge Engineering*. 68(11): 1271-1288.

[28] Shannon, C. E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*. 27: 379-423, 623–656.

[29] Apostolopoulos, T., and Vlachos, A. 2011. Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem. *International Journal of Combinatorics*. 2011: 523806.

[30] Hassanzadeh, T., Vojodi, H., and Moghadam, A. M. E. 2011. An Image Segmentation Approach Based on Maximum Variance Intra-Cluster Method and Firefly Algorithm. *IEEE Seventh International Conference on Natural Computation (ICNC), Shanghai*. 1817-1821.

[31] Bojic, I., Podobnik, V., Ljubi, I., Jezic, G. and Kusek, M. 2012. A Self-Optimizing Mobile Network: Auto-Tuning the Network with Firefly-Synchronized Agents. *Elsevier, Information Sciences*. 182(1): 77-92.

[32] Hassanzadeh, T., Faez, K., and Seyfi, G. 2012. A Speech Recognition System Based on Structure Equivalent Fuzzy Neural Network Trained by Firefly Algorithm. *IEEE International Conference on Biomedical Engineering (ICoBE)*. 63-67.

[33] Tan, S. C. 2012. Simplifying and Improving Swarm Based Clustering. IEEE Congress on Evolutionary Computation (CEC), Brisbane, QLD. 1-8.

[34] Bonabeau, E., Dorigo, M., and Theraulaz, G. x. 1994. *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity.

[35] Mohammed, A. J., Yusof, Y., and Husni, H. 2015. Document Clustering Based on Firefly Algorithm. *Journal of Computer Science*. 11(3): 453-465.

[36] Mohammed, A. J., Yusof, Y., and Husni, H. 2016. Discovering Optimal Clusters using Firefly Algorithm. *International Journal of Data Mining, Modeling and Management*. 8(4): 330-347.

[37] Mohammed, A. J., Yusof, Y. and Husni, H. 2016. Integrated Bisect K-means and firefly Algorithm for Hierarchical Text Clustering. *Journal of Engineering and Applied Sciences*. 11(3): 522-527. ISSN 1816-949X.