International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Hybrid Approach for Web Document Clustering Using K-means and Artificial Bee Colony Algorithm

M.M. Gowthul Alam[1*], S. Baulkani[2]

[1]*Department of Computer Science and Engineering, National College of Engineering, Tirunelveli, Tamilnadu, India*
[2]*Department of Electronics and Communication, Government College of Engineering, Tirunelveli, Tamilnadu, India*
* Corresponding author's Email: gowthulalam2005@gmail.com

Abstract- Nowadays data growth is directly proportional to time and it is a major challenge to store the data in an organised fashion. Document clustering is the solution for organising relevant documents together. In this paper, a web clustering algorithm namely WDC-KABC is proposed to cluster the web documents effectively. The proposed algorithm uses the features of both K-means and Artificial Bee Colony (ABC) clustering algorithm. In this paper, ABC algorithm is employed as the global search optimizer and K-means is used for refining the solutions. Thus, the quality of the cluster is improved. The performance of WDC-KABC is analysed with four different datasets (webkb, wap, rec0 and 7sectors). The proposed algorithm is compared with existing algorithms such as K-means, Particle Swarm Optimization, Hybrid of Particle Swarm Optimization and K-means and Ant Colony Optimization. The experimental results of WDC-KABC are satisfactory, in terms of precision, recall, f-measure, accuracy and error rate.

Keywords: *K-Means, artificial bee colony algorithm, Web document Clustering*

1. Introduction

Today's world revolves around data and the massive storehouse of data is the internet. Data growth is directly proportional to time and the stored data must be managed properly. Every day the internet deals with several petabytes (PB) of data. Thus, to extract the useful information from the voluminous database is the major challenge for the researchers. Most of the information is in the form of digital documents. Hence a mechanism is needed to systematize the documents, such that the end users are able to retrieve the relevant data in a reasonable period of time. Document clustering is an effective mechanism for clustering related documents together. This clustering approach provides a way for finding the relevant documents in minimal period of time.

The literal meaning of clustering is grouping; thus document clustering is systematizing the documents into several classes based on the degree of relevance. Each class is denoted as a cluster and the entities within a cluster are closely related to each other. On the other hand, the entities of two different clusters will appear different [1-3]. Some of the main application areas of document clustering are data mining [4] and Content based Information Retrieval (CBIR) [5-8].

The main goal of a web document clustering algorithm is (1) to produce appropriate clusters for the end user, (2) to assign the available documents to the most relevant cluster, (3) to respond the end user instantly. An alternative method for information retrieval is document clustering based on web in organizing the information. Clustering based on fuzzy technique enhances the search faster and focuses on appropriate and meaningful results. This method overcomes the dependency of large training corpus size of information. K-means clustering is sensitive to form the initial centroid and the number of clusters. Artificial Bee Colony optimization is the latest optimization techniques, which is independent of previous knowledge and

number of clusters available. The main advantage of this method is that it does not need initial parameterization.

In this paper, a new algorithm namely 'WDC-KABC' is proposed to cluster web documents. The proposed algorithm is the combination of K-means and Artificial Bee Colony (ABC) algorithm. The reasons for incorporating K-means algorithm are its simplicity and efficiency [9]. Initially, ABC algorithm is employed to achieve clustering [10] which is followed by the application of K-means algorithm. The initial cluster centre is fixed by ABC algorithm. Through experimental analysis, it is proved that the performance of WDC-KABC is better than the other comparative algorithms, in terms of execution time, F-measure, precision and recall.

The remainder of this paper is classified as follows. Section 2 summarizes the existing works related to the web document clustering. Section 3 describes the background information about ABC and K-means algorithm. The proposed web document clustering algorithm (WDC-KABC) is presented in section 4. Section 5 analyses the performance of the proposed system with the existing techniques. Finally, section 6 is loaded with the concluding remarks.

2. Review of Literature

Basically, document clustering algorithms can be categorised into three different types. They are data-centric algorithms, description-aware and description-centric algorithms [32]. Among the three categories, data-centric algorithms are popular and its variations are based on partitions, density, fuzzy and hierarchical [32-36]. The mostly employed algorithms are based on hierarchy and partitions [34].

Hierarchical clustering algorithms are based on trees and the relevant entities are grouped together based on similarity measures. The best example is the Un-weighted Pair-Group Method using Arithmetic average (UPGMA) [35, 37]. The major drawback of the algorithm is the time consumption. The hierarchical algorithms are further divided into two types. They are agglomerative and divisive algorithms [38]. Agglomerative algorithms assign every document to cluster randomly and then join the clusters iteratively. On the other hand, divisive algorithms divide the entire documents into some clusters and the number of clusters will be improved.

Partition based algorithms decompose the data into clusters and transfer the entities between clusters based on an objective function [13, 15]. The K-means is the best example for this type of algorithm. The complexity of this algorithm is very less and is very fast. However, this algorithm tries hard to select initial cluster centres but it is not suitable for large set of data. This traditional partition based algorithms are focussed to get improve by means of optimization techniques. The global objective function can be selected by the optimization algorithms such as Genetic algorithm [17, 18], Ant Colony Optimization [19, 20], Particle Swarm Optimization [16]. Partition based algorithms are propitious than hierarchical algorithms. The main reason is hierarchical algorithms consume more time than partition based algorithms.

Motivated by these algorithms, the proposed work strives to provide a partition based web document clustering algorithm, which improves the functionality of K-means and by increases Artificial Bee Colony (ABC) algorithm. The experimental results of the proposed algorithm are satisfactory. The preliminaries of the algorithm are presented in section 3.1 and 3.2.

3. Background information

3.1 Artificial Bee Colony (ABC) Algorithm

ABC is an optimization algorithm, which imitates the real acts of honey bees, proposed in [10]. The most important components of ABC algorithm are its food source, employed and unemployed bees. The main theme of this algorithm is to arrive at the best food source. The standard pseudo code of the ABC algorithm is presented in algorithm 1.

The most basic ABC algorithm consists of three phases. They are initialization, employed, onlooker and scout bees phase. Each phase is replayed until the maximum count of iterations is reached. In the initial phase, the count of solutions and the control parameters are fixed. The employed bees phase deals with the search of new high quality food sources in the nearby locality of old food source. The new food source is then evaluated for its fitness, which is then followed by the comparison of the old and the new food source by means of greedy selection. The collected knowledge about the food source is distributed among the onlooker bees present in the beehive.

In the next phase, the onlooker bees follow a probabilistic approach to select the food sources with respect to the information provided by the employed bees. This is followed by the calculation of the fitness function of the food source, which is located nearby the selected food source. Finally, the

old and the new food sources are compared by the greedy selection.

In the final phase, the employed bees turn to scout bees, when their solutions cannot be enhanced within a predefined count of iterations. The solutions so found by the bees are dropped out. At this point, the scout bees search for new food source again. Using this functionality, the poor solutions are dropped out. These three phases continue its process until the stopping point is reached [11, 12].

Algorithm 1

1: Input: Training data;
2: Produce initial population $i_p = 1$ *to MC*
3: Calculate the fitness function of the population
4: Fix counter=1
5: Do

// Employed bees phase

6: Search for the food source;
7: Calculate the fitness function;
8: Employ greedy selection process;
9: Compute the probability for the food source;

// Onlooker bees phase

10: Select food source based on the probability values;
11: Generate new food source;
12: Calculate the fitness function;
13: Apply greedy selection process;

// Scout bees phase

14: If food source drops out then swap it with new food source;
15: Save the best food source;
16: Counter + =1;
17: While counter=MC;

3.2 The K-means Algorithm

The K-means algorithm is an effective algorithm that is employed for the clustering purpose. The algorithm is well known for its simplicity and its ease of implementation. The idea of this algorithm is to cluster the data points. The K-means algorithm depends on the Sum Of Square Error (SOSE) and this can be given by

$$\mathrm{SOSE} = \sum_{x=1}^{cc} \sum_{y=1}^{c} B_{yx} \| k_y - l_x \|^2 \qquad (1)$$

The K-means algorithm is partitioned with the count of clusters and the objects needed to be clustered. The membership of object id determined and the objects are rearranged. This is followed by the renewal of cluster points and this process continues until a stopping criterion is met.

4. Proposed Algorithm: WDC-KABC

Web document clustering involves the representation of a data model, similarity measure selection, clustering model and the validation [21]. This work sticks on to the aforementioned point and are presented in the forthcoming subsections. The proposed algorithm is explained in Fig. 1.

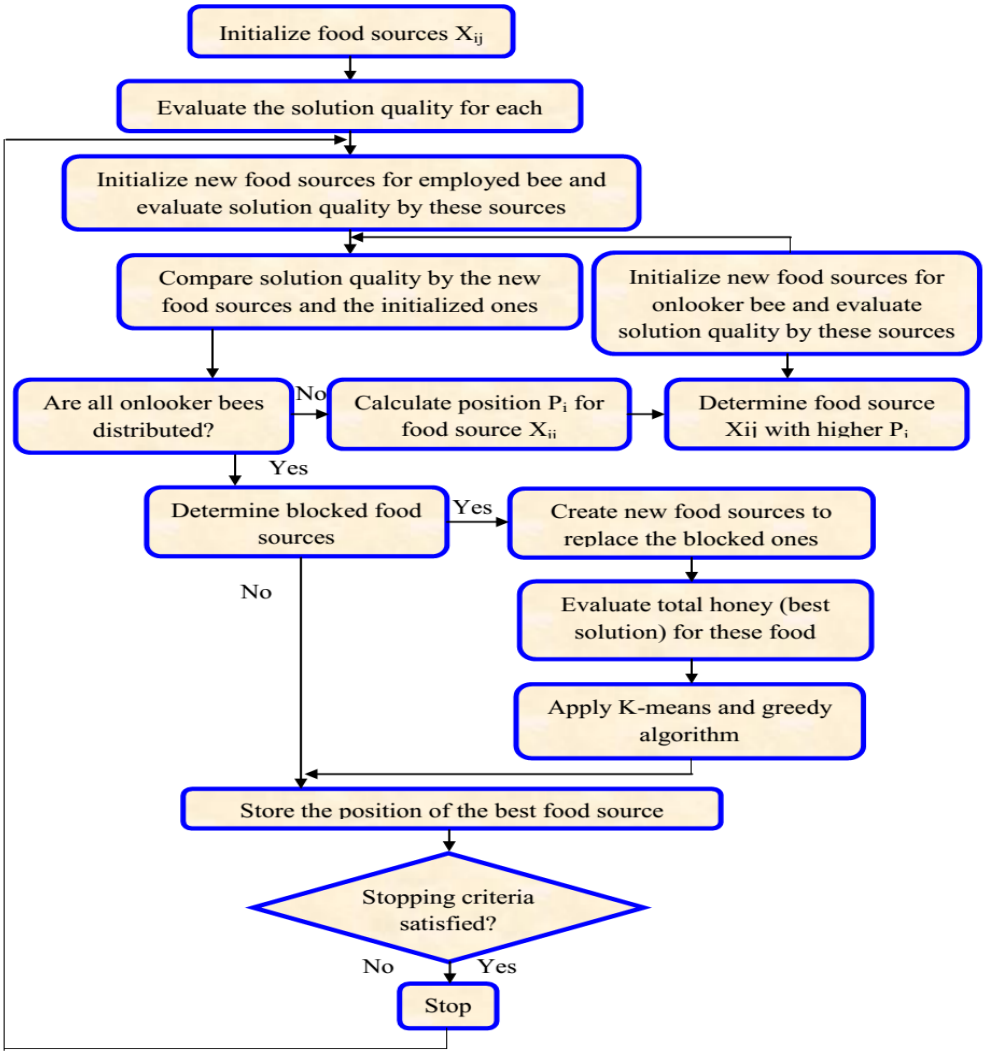

Figure.1 Proposed Architecture Diagram

4.1 Document Representation

Documents can be represented by various models. However, the most frequently employed model for document representation is Vector Space Model (VSM). Vector Space Model represents the text documents as vectors [5, 22]. Let $Doc=(Doc_1,Doc_2,Doc_3,....Doc_i)$ be the accumulation of documents and TS be the term set of the documents Doc, where $TS=(T_1,T_2,....T_K)$ In this work, every document is defined as a point in a k dimensional vector space. Thus, the dimension is based on term set of the document. Every dimension denotes a different term $Doc_m= (Wt_{m1}, Wt_{m2}, Wt_{m3}, ... Wt_{mk})$. Where, the value of m ranges between 1 to i. Each element of Doc_m is the weight of the terms available in the document. Thus, Doc_m is based on the level of association between the term and the document. Measurement of the level of association between the term and the document is referred to as term weighting process. Term frequency-inverse document frequency model is the famous model to determine the level of association between the term and the document. This model focuses on the local and the global occurrence frequency of the term in the documents and is presented in equation (2).

$$wt_{my} = i_{my} \times i.d.f \qquad (2)$$

Where, i_{my} is the occurrence frequency of the term y in m^{th} document and $i.d.f$ is given by equation (3).

$$i.d.f = \log\left(\frac{i}{i_y}\right) \qquad (3)$$

Inverse document frequency $(i.d.f)$ determines the occurrence frequency of term y in a group of documents. In case of the presence of a

term in all the documents, the term cannot play the role of a keyword. For instance, when the term frequency of a word equals the total number of documents, then the weight assigned to that term becomes 0. Mostly, such terms can be articles, connectors, prepositions and pronouns. These terms are not considered to be significant in this work and so are eliminated.

4.2 Similarity Measure

The process of clustering completely depends on the similarity of terms or documents, as similar entities can alone be grouped or clustered. This step is the predecessor of the clustering process. The similarity measure determines the level of association between the documents. There are several performance similarity measures such as Euclidean distance, cosine similarity, Jaccard coefficient and Pearson correlation coefficient [23, 24]. This work proposes to incorporate cosine similarity measure because of its wider range of applications in text mining. The main objective of a similarity measure is to obtain the degree of association between two documents. This is achieved by the calculation of cosine angle between two different vectors and can be calculated by Equation (4).

$$S_m(Doc_m, Doc_q) = \cos(Doc_m, Doc_q) \quad (4)$$

$$\cos(Doc_m, Doc_q) = \frac{\sum_{y=1}^{k} wt_{my} wt_{qy}}{\sqrt{\sum_{y=1}^{k} wt_{my}^2 \cdot \sum_{y=1}^{k} wt_{qy}^2}}, m \quad (5)$$

The outcome of the above presented equation is the similarity between two documents and the value is within the range between 0 and 1.

4.3 Clustering algorithm

A combination of Artificial Bee Colony and k-means algorithm is proposed for clustering the web documents. ABC colony algorithm is an efficient population based optimization algorithm and it imitates the behaviour of real bees. The k-means algorithm is efficient and fast, however the problem is on finding initial cluster point. This work proposes to locate the initial cluster point with the help of bees and these clusters are refined by the k-means algorithm. We propose to combine both ABC and k-means algorithm, so as to inherit the merits of both the algorithms. ABC is efficient but consumes more time for convergence. The k-means algorithm is also known for its faster convergence but struggles in locating the initial cluster point. Thus, a new algorithm is presented for improving the efficiency and reducing the execution time. The steps involved in the proposed algorithm are explained below.

Step 1: Parameters initialization

The parameters that need to be initialized are the maximum count of iterations or the maximum time can be given in milliseconds, position of the food source (cluster centre) and occurrence frequency threshold of the terms for labelling clusters. The fitness value for this algorithm is the degree of relevance between two documents. Thus, the initial population of food sources are distributed randomly.

Step 2: Document pre-processing

This step is concerned with the removal of articles, connectors, prepositions and pronouns. This step is to weed out the unwanted terms, so as to make the clustering process effective. The stop words are eliminated [27]. Some of the sample stop words which were removed from the documents are presented in table 1.

Table 1. Sample stop words

Sample stop words			
a	an	the	about
above	according	despite	along
again	among	apart	around
before	after	between	but
by	because	with	without
over	under	near	on
of	till	until	in
below	behind	beside	beyond
you	me	I	we

Let $Doc = (Doc_1, Doc_2, Doc_3 ... Doc_i)$ be the accumulation of documents and TS be the term set of the documents Doc, where $TS = (T_1, T_2, ... T_K)$. In this work, every document is defined as a point in a k dimensional vector space. Thus, the dimension is based on the term set of the

document. Every dimension denotes a different term and it is $Doc_m = (wt_{m1}, wt_{m2}, wt_{m3} ... wt_{mk})$, where the value of m ranges between 1 to i. Each element of Doc_m is the weight of the terms present in the document. Thus, Doc_m is based on the level of association between the term and the document. Similarity between the documents or the similarity between the document and the cluster centre is calculated by equation (2).

Step 3: The execution of the proposed algorithm depends on the source of food, which are the solutions. Let food source $fs = 1,2,..i$ is a k dimensional vector, where k is the multiplicative result of documents and the cluster size. The initial count of documents in a cluster is 2 and the maximum number of documents to be in a cluster is 8.

Step 4: In this step, the fitness of the population is calculated by the Equation (6) and is given below.

$$f_i = \sum_{c=1}^{n} \sum_{doc_s \in CCi} ||doc_s - CCi||^2 \qquad (6)$$

Where, c is the cluster, CC is the cluster centre and doc is the document. The above equation determines the distance between the document and the cluster centre. The main objective is to have minimal fitness value.

Step 5: After finding the fitness of the population, the employed bees search for the new food source in its neighbourhood and provides a new food source from its locality. This new food source is tested for its fitness by the K-means algorithm and the greedy selection is applied. In case, if the degree of similarity (fitness) of the new document with the cluster centre is more than the similarity between the old document and the cluster centre, then its memory is loaded with the new document and is computed by Equation (7). By this way, the employed bee search for the better documents with respect to the cluster centre. This is followed by the calculation of probability of the food source and is calculated by Equation (8).

$$new_{i,j} = a_{i,j} + \partial(a_{i,j} - a_{k,j}) \qquad (7)$$

The employed bees search for the new documents with better degree of relevance with respect to the cluster centre, in its neighbourhood.

In (7), $a_{i,j}$ is the location of the initial document and is stored in memory of the employed bee, $a_{k,j}$ is the new document and ∂ is in the range of [1, -1]. The new document is found by changing a dimension over a. By this way, the employed bee moves in its neighbourhood and the location bound is reset.

$$prob_i = \frac{f_i}{\sum_{n=1}^{fs} fn} \qquad (8)$$

Where, f_i is the fitness of the i^{th} document and n is the total number of cluster centres. Thus, the probability is calculated.

Step 6: The onlooker bee, which is waiting on the bee-hive, chooses the document based on the so calculated probability value and tries to find a document in the neighbourhood. If a document is found in its neighbourhood, then the degree of relevance is found. This is followed by the application of K-means and the greedy selection, as in step 6. This notifies that any number of onlooker bees can probabilistically select a single document with high fitness value. Finally, the obtained best solution is saved. This process continues till the stopping point for execution is reached. The overall structure of proposed document clustering algorithm is presented in algorithm 2.

Algorithm 2

1. Initialize the algorithm parameters
2. Pre-process the document
3. Randomly assign the population of food sources
4. Determine the fitness of the population by (6)
5. While
6. For each employed bee
7. Produce new food source;
8. Calculate fitness of the food source;
9. Employ K-means and greedy selection;
10. Calculate the probability of food source by (8);
11. For each onlooker bee
12. Choose the food source with respect to step 1.
13. Produce new food source and compute its fitness by (6);
14. Apply K-means and greedy selection
15. Compare and swap the solutions if new source is better;
16. Save the best food source;
17. end while (termination condition not met);

5. Experimental results and discussion

In this section, the performance of the proposed algorithm is verified by comparative analysis made with the existing algorithms such as PSO, KPSO, K-means, ABC, and KABC. The datasets which are exploited for the experimentation process are webkb, wap, 7sectors and re0. These are the benchmark datasets for document clustering [23, 25, 26]. The summary of the exploited datasets is presented in table 2.

5.1 Dataset description

Webkb: This dataset consists of 8282 documents, which are categorized into seven categories. They are student, faculty, staff, department, course, project and others. Webkb can be downloaded from [28].

fbis: This dataset is a part of TREC-5 collection and can be downloaded from [29]. This dataset contains 2463 documents with 17 classes.

Re0: This dataset is a part of Reuters-21578 text categorization and contains 1504 documents with 13 classes and can be downloaded from [30].

7sectors: This dataset contains 4581 documents with six different classes. This dataset can be downloaded from [31].

Table 2. Exploited datasets

Data sets	Document Count	Class count	Words Count	
			Before Pre-processing	After Pre-processing
Webkb	8282	7	20682	2986
Fbis	2463	17	12764	1530
Re0	1504	13	2886	520
7sectors	4582	6	10863	1260

5.2 Performance analysis

The performance metrics used to measure the effectiveness of the proposed algorithm are precision, recall, F-measure and accuracy. Precision and recall are calculated for all the documents that are grouped under a cluster and checked with the real class label.

Precision: Precision is the count of documents with a label i, but misclassified by the clustering algorithm as the document is with label j to the count of documents with label j. This is computed by Equation (9).

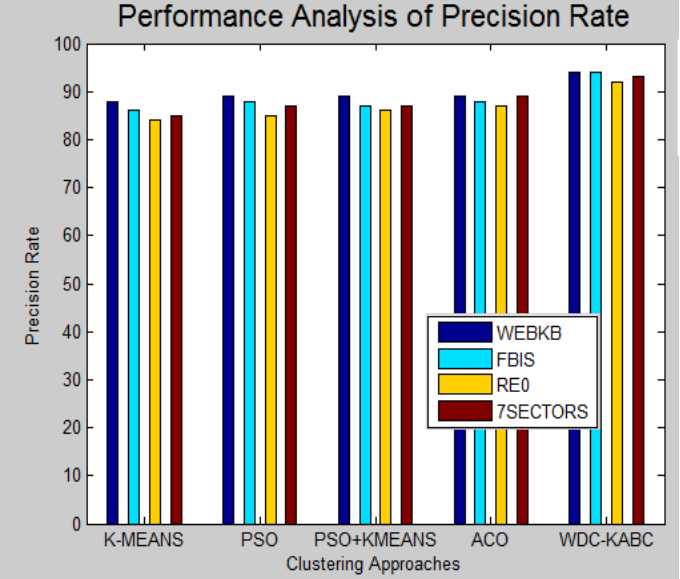

Figure.2 Comparative analysis of precision rate with Webkb, FBIS, Re0 and 7Sectors

$$\Pr ec(i,j) = \frac{doc_{ij}}{doc_j} \tag{9}$$

Where, doc_{ij} is the total count of documents which are really labelled under i, but misclassified as j by the employed clustering algorithm. doc_j is the count of documents that are grouped under label j. The precision rates of the compared algorithms for different datasets are depicted in Fig. 2.

From the experimental results, it is evident that the precision rate of WDC-KABC is good when compared with K-means, PSO, PSO+K-means and ACO. The precision rate of WDC-KABC ranges from 95.82 to 98.18. The results assure that the documents clustered together are relevant out of the total count of clustered documents while comparing with other existing methods.

Recall: Recall is the number of documents, which are grouped under i in reality but are misclassified as the document under label j. This is measured by the following Equation.

$$\mathrm{Re}c(i,j) = \frac{doc_{ij}}{doc_i} \tag{10}$$

Where, doc_{ij} is the total count of documents which are really labelled under i, but misclassified as j by the employed clustering algorithm. doc_1 is the count of documents that are grouped under label i.

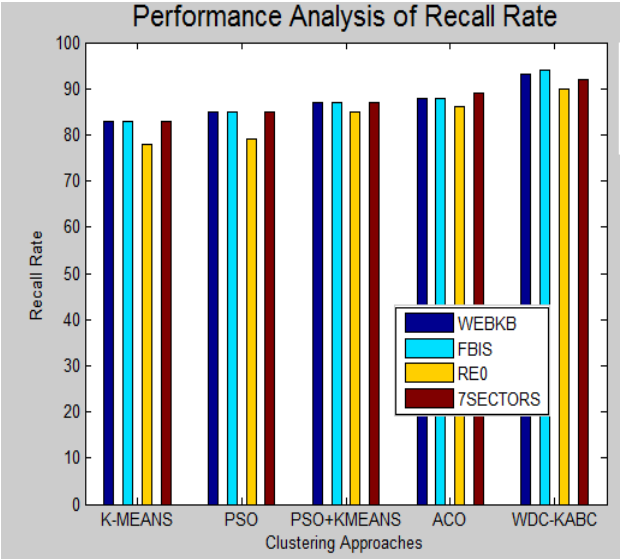

Figure.3 Comparative analysis of recall rate with Webkb, FBIS, Re0 and 7Sectors

The comparative analysis of the recall rates is carried out by considering four different datasets and the results are depicted in Fig. 3. On analysing the experimental results of recall rate with respect to different datasets, it is proved that the proposed algorithm shows good recall rate. The recall rate of WDC-KABC lies between 92.19 to 97.36. Thus, the recall rate shows that the WDC-KABC clusters the relevant documents together out of the total count of actual relevant documents.

F-measure: F-measure is the complete clustering result of the documents that is based on the precision and recall and it is calculated by Equation (11). This measure calculates the overall sum of individual cluster's F-measures.

$$Fm = \sum_i \frac{doc_i}{doc\_count} \max_j (F_m(i,j)) \qquad (11)$$

Where $F_m(i,j)$ is the F-measure of individual clusters and is calculated by (12)

$$F_m(i,j) = \frac{2 \times prec(i,j) \times rec(i,j)}{prec(i,j) + rec(i,j)} \qquad (12)$$

The value of F-measure is directly proportional to the quality of the cluster solution. The proposed algorithm shows greater F-measure than the analogous algorithms and is evident in the experimental analysis as shown in Fig. 4.

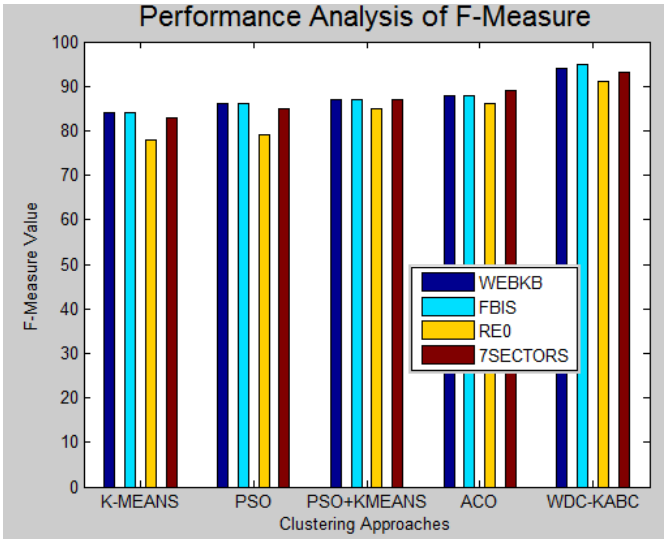

Figure.4 Comparative analysis of F-Measure with Webkb, FBIS, Re0 and 7Sectors

On analysing the results, the proposed algorithm outperforms all the existing algorithms with higher rates of F-measure. The quality of clusters is directly proportional to the value of F-measure. The F-measure of the proposed algorithm ranges between 92.97 and 98.17. Thus, the quality of clusters is found to be satisfactory in the proposed algorithm.

Accuracy: The accuracy rate depends on the effectiveness of the algorithm in clustering related documents together. The accuracy rate of the proposed algorithm is comparatively improved than other algorithms.

$$accuracy = \frac{c \cdot a \cdot d + c \cdot r \cdot d}{clustered\ documents} \qquad (13)$$

Where, $c.a.d$ is the correctly accepted document and $c.r.d$ is the correctly rejected document by the clustering algorithm. The accuracy rate of the proposed algorithm is analysed with several datasets and the results are shown in Fig. 5.

The main objective of a clustering algorithm is to increase the accuracy rate of clustering, by means of grouping relevant documents together. The range of accuracy rates being shown by the proposed algorithm is between 96 to 98 percentages and is found to be satisfactory.

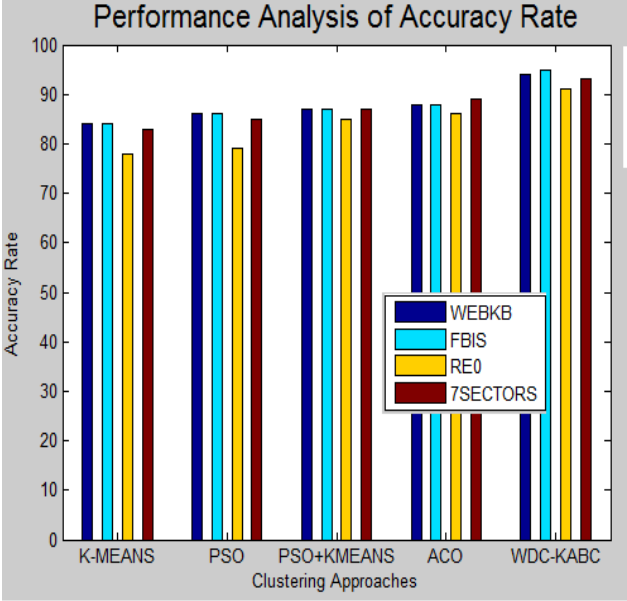

Figure.5 Comparative analysis of accuracy with Webkb, FBIS, Re0 and 7Sectors

Misclassification Rate: Misclassification rate (*mr*) is the count of documents that were wrongly clustered out of the total count of clustered documents and is calculated by Equation (14).

$$mr = 100 - accuracy \tag{14}$$

Misclassification rate in percentage is obtained by subtracting the accuracy rate from 100. The clustering algorithm is found to be good with a minimal misclassification rate. The proposed algorithm is tested for misclassification rate and the results are showed in Fig. 6.

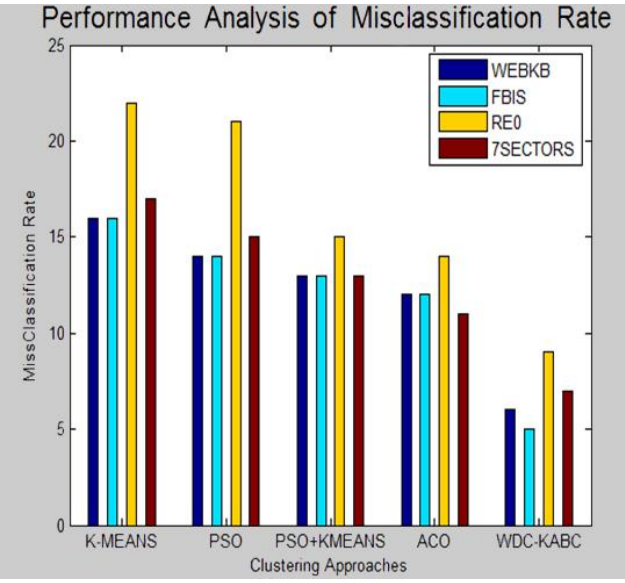

Figure.6 Comparative analysis of misclassification rate with Webkb, FBIS, Re0 and 7Sectors

algorithm is between 2 and 4, which makes the results more accurate. Contemporary, IR system gives millions of documents at single search. It is tiresome to go through all the documents when the query is small and ambiguous, the user go throughs only few first pages. And the first few pages will be the dominating documents of frequently searched by the user. So a need for good web search system is essential and is fulfilled by our proposed method. Based on the experimental results, it is evident that the precision rate, recall, F-measure, Accuracy and misclassification produces good results. K-mean clustering is computationally faster than the other methods which are very much needed in document search K-mean clustering produces tighter clusters than the other clustering methods especially if the clusters are globular.

6. Conclusion

This paper proposed a web document clustering algorithm namely WDC-KABC, which is a combination of K-means and Artificial Bee Colony (ABC) algorithm. The main purpose of clustering was to group relevant documents together, such that the degree of relevance between the documents in a cluster is more and the documents between two clusters show lesser degree of relevance. ABC algorithm was used as the global search optimizer and K-means is employed as the local solution optimizer. This work employed four different datasets for checking the performance of WDC-KABC. The experimental results of WDC-KABC were satisfactory, when it was compared with the several algorithms. In future, the quality of clusters can further be improved by designing a new bio-inspired algorithm. Gray Wolf Optimizer (GWO) algorithm can be used to improve the clustering methods.

References

[1] Das, Swagatam, and Amit Konar. "Automatic image pixel clustering with an improved differential evolution", *Applied Soft Computing* 9.1, 226-236, 2009.

[2] Francesco Adamo, Filippo Attivissimo, Attilio Di Nisio, Maurizio Spadavecchia, "An automatic document processing system for medical data extraction,"Measurement 61 (2015) 88–99.

[3] Swagatam, Ajith Abraham, and Amit Konar. "Automatic kernel clustering with a multi-elitist

particle swarm optimization algorithm", *Pattern recognition letters* 29.5, 688-699, 2008.

[4] Jiawei, Han, and Micheline Kamber. "Data mining: concepts and techniques", *San Francisco, CA, itd: Morgan Kaufmann*, 2006.

[5] Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. "Modern Information Retrieval*", Addision Wesley*. Vol. 463. New York: ACM Press, 1999.

[6] Hammouda, M Khaled and Mohamed S. Kamel, "Efficient phrase-based document indexing for web document clustering", *IEEE Transactions on knowledge and data engineering* 16.10, 1279-1296, 2004.

[7] Kalashnikov, D. V., Chen, Z., Mehrotra, S., & Nuray-Turan, R. (2008). Web people search via connection analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20(11), 1550-1565.

[8] Aggarwal, C.Charu and Chandan K. Reddy, "Data clustering: algorithms and application*", CRC Press*, 2013.

[9] MacQueen, James, "Some methods for classification and analysis of multivariate observations", *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. No. 14. 1967.

[10] Karaboga and Dervis, "An idea based on honey bee swarm for numerical optimization", Vol. 200. Technical report-tr06, *Erciyes University, engineering faculty, computer engineering department*, 2005.

[11] D.Karaboga, B. Gorkemli, C Ozturk., and N. Karaboga. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review, 42(1), 21-57.

[12] Karaboga, Dervis and Celal Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm." *Applied soft computing* 11.1, 652-657, 2011.

[13] Jain, Anil K., M. Narasimha Murty and Patrick J. Flynn, "Data clustering: a review", ACM *computing surveys (CSUR)* 31.3, 264-323, 1999.

[14] C Manning, P Raghavan and H Schütze, "Introduction to Information Retrieval", *Cambridge University Press*, Cambridge, England, 2008.

[15] Reddy, Damodar, Prasanta K Jana, "Initialization for K-means clustering using Voronoi diagram", *Procedia Technology* 4, 395-400, 2012.

[16] X Cui, T.E Potok, P Palathingal, "Document clustering using particle swarm optimization," in: *Proceedings of the IEEE swarm intelligence symposium, SIS, Springer, Piscataway, IEEE Press*, pp.185 – 191, 2005.

[17] V.J Rayward-Smith, "Meta heuristics for clustering in KDD", *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 3, IEEE Press, Piscataway, pp. 2380–2387, 2005.

[18] Cole R.M, "Clustering with Genetic Algorithms", *Master's thesis, Department of Computer Science, University of Western Australia, Australia*, 1998.

[19] Saatchi S, Hung C.C, "Hybridization of the ant colony optimization with the K-means algorithm for clustering", *Lecture Notes in Computer Science, Image Analysis*, vol. 3540, Springer, Berlin, 2005.

[20] G Salton, "Automatic Text Processing", *Addison-Wesley*, 1989.

[21] Aliguliyev, M Ramiz, "Performance evaluation of density-based clustering methods", *Information Sciences* 179.20, 3583-3602, 2009.

[22] Veritt. B, "Cluster Analysis", *Second Edition, Halsted Press, New York*, 1980.

[23] Anna Huang, "Similarity Measures for Text Document Clustering", *Proceedings of the New Zealand Computer Science Research Student Conference*, pp 49-56, 2008.

[24] T Korenius, Laurikkala and M Juhola, "Principal component analysis." *cosine and Euclidean measures in information retrieval, Information Science* 177, 4893–4905, 2007.

[25] T Li, C Ding, "Weighted consensus clustering, in: Proceedings of the SIAM International Conference on Data Mining", (SDM 2008), *Atlanta, USA*, pp. 798 – 809, 2008.

[26] Zhao Y, Karypis G, "Empirical and theoretical comparisons of selected criterion functions for document clustering", *Machine Learning* 55, 311–331, 2004.

[27] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

[28] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

[29] http://trec.nist.gov

[30] www.daviddlewis.com/resources/testcollections/reuters21578

[31] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/bootstrappingIE/7sectors.tar.gz

[32] C Carpineto, S Osinski, G Romano, D Weiss, "A survey of Web clustering engines", *ACM Computing Survey* 41, 1–38, 2009.

[33] P Berkhin, J Kogan, C Nicholas, Teboulle M, "A survey of clustering data mining techniques, in: S. Sri (Ed.), *Grouping Multidimensional Data, Springer-Verlag*, pp. 25–71, 2006.

[34] K Hammouda, "Web Mining: Clustering Web Documents A Preliminary Review", pp. 1–13, 2001.

[35] Jain A.K, and Dubes R.C, "Algorithms for clustering data", *Prentice-Hall, Inc*., 1988.

[36] Steinbach M, Karypis G, Kumar V, A "Comparison of document clustering techniques, in: KDD workshop on text mining", *ACM Boston, MA, USA*, pp. 1–20, 2000

[37] Y Li, S.M Chung, J.D Holt, "Text document clustering based on frequent word meaning sequences", *Data Knowl. Eng*. 64, 381–404, 2008.

[38] S Xu, J Zhang, "A parallel hybrid web document clustering algorithm and its performance study", *The Journal of Supercomputing* 30, 117–131, 2004.