Clustering Algorithm Comparison of Search Results Documents

David, Raymondus Raymond Kosala
[1]STMIK Pontianak, Pontianak, Indonesia
[2]Universitas Bina Nusantara, Jakarta, Indonesia
DavidLiauw@gmail.com, RKosala@binus.edu

***Abstract*- Document clustering is one of the popular studies of data mining. This research focused on creation of the application system of document clustering of search results documents through clustering algorithms of Ant Colony Optimization, Forgy and ISODATA. Created applications were used to group and ease search results documents. Clustered documents were articles of journals, theses, thesis proposals, and ebooks. Indexing and searching the documents apply Apache Lucene, the search engine. Ant Colony Optimization algorithm was compared to partitioning clustering of Forgy and ISODATA. Comparison was on examination of processing time of clustering, variance, and the sum of squared errors. Experiments of groups of documents and datasets were conducted. To conclude, clustering results of the three methods show identical variance and produce high intraclass similarity and low interclass similarity. Also, in comparison to others, clustering through algorithm of Ant Colony Optimization takes the most time.**

Keywords— Document Clustering, Ant Colony Optimization, Forgy, ISODATA

I. INTRODUCTION

Documents in libraries like articles of journals, theses, books, and thesis proposals are usually not stored based on the content. Hence, difficulties of searching them appear. Categorizing them into certain classes is also difficult because of the needs of reading and understanding the content. After reading the whole content, the classes are determined and division of documents are made.

In a scientific study named Information Retrieval, several methods were proposed to ease the search of information of a huge number of digital documents [1]. One of them is clustering, i.e. classification of similar text-based data or documents. Clustered documents create other clusters supporting the classification. The search of documents without clustering can show certain keywords. This is applied by the search engines like Google or Yahoo.

Therefore, system enabling classification and easing the search of documents is needed. There are numerous applicable clustering methods. However, the authors only concentrated on the algorithm of Ant Colony Optimization. The reason is that it was rare to find it performing as a partitioning clustering method in the previous research [2,3]. Such the algorithm would further be compared to the other partitioning clustering methods, namely Forgy [4] and ISODATA [5] for document clustering. The documents were articles of journals, theses, thesis proposals, and ebooks.

A clustering method would be combined with the weighting of TF-IDF. This becomes an automatic solution to classification of unstructured data like documents. It should be noted that TF-IDF itself is a popular method and has quite accurate calculation [6].

Based on above description, the authors were interested in implementing and comparing the three clustering algorithms such as Ant Colony Optimization, Forgy and ISODATA in the searching process of documents. Queried documents with keywords were grouped into a number of clusters. The system received inputs of documents searched. The related scores among desired words were calculated afterwards. The higher they are, the more specific the clusters are.

An expected result was that searching process would be more efficient in comparison to the direct search of collections of documents without causing the quality of search results. Comparison of the three algorithms could be seen based on processing time of clustering, variance, and the sum of squared errors.

II. RESEARCH METHODS

The research was conducted based on these steps: planning, analysis, implementation designs, examination, and discussion by using the three clustering algorithms such as Ant Colony Optimization, Forgy and ISODATA.

Ant Colony Optimization

The problem of data clustering is modeled as the clustering optimization problem. The set of data consisting of Object Data *m* with Attribute *n* is given. A number of clusters (*g*) are determined. Equation (1) indicates objective functions [7].

$$J(W,C) = \sum_{i=1}^{m}\sum_{j=1}^{g} w_{ij}\left\|X_i - C_j\right\| \quad \text{..... (1)}$$

Where $\left\|X_i - C_j\right\| = \sqrt{\sum_{v=1}^{n}(x_{iv} - c_{jv})^2}$ and

$$\sum_{j=1}^{g} w_{ij} = 1, \quad i = 1,...,m$$

If Data *i* is included in Cluster *j*, $w_{ij} = 1$, if not $w_{ij} = 0$.

$$C_j = \frac{\sum_{i=1}^{m} w_{ij} X_i}{\sum_{i=1}^{m} w_{ij}}, j = 1,...,g \quad \text{.....(2)}$$

Notes:

x_i : The -*i* object data vector and $x_i \in R^n$

x_{iv} : The -*v* attribute score of the -*i* object data

c_j : The vector of the -*j* cluster centroid vector and $c_j \in R^n$

c_{jv} : The score of the -*v* attribute from the -*j* cluster centroid

w_{ij} : The combined score of x_i and c_j

X : An $m \times n$ data matrix

C : A $g \times n$ cluster centroid matrix

W : An $m \times g$ weight matrix

The trace of each ant colony is represented in a pheromone matrix. It is normalized by using the following equation [8]:

$$P_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^{g} \tau_{ik}}, j = 1,...,g \quad \text{.....(3)}$$

P_{ij} is a pheromone normalization probability matrix of Element *i* on Cluster *j*. The distance between Object *i* and Cluster *j* from Ant *k* ($d_k(i,j)$) can be defined in the following equation [7]:

$$d^k(i,j) = \sqrt{\sum_{v=1}^{n} (x_{iv} - c_{jv}^k)^2} \quad \text{.....(4)}$$

Cluster *j* of each ant is selected by using two strategies such as exploitation and exploration. Equation of exploitation is as follows [9][10]:

$$j = \begin{cases} \arg\max_{u \in N_i} \{[\tau(i,u)]^\alpha [\eta^k(i,u)]^\beta\} & if \quad q \le q_0 \\ P^k(i,j) & otherwise \end{cases}$$

.....(5)

And equation of exploration is as follows [9][10]:

$$P^k(i,j) = \frac{[\tau(i,u)]^\alpha [\eta^k(i,u)]^\beta}{\sum_{j=1}^{g} [\tau(i,u)]^\alpha [\eta^k(i,u)]^\beta} \quad (6)$$

Where score η_{ij}^k is obtained through the following equation [9]: $\eta_{ij}^k = \frac{1}{d^k(i,j)}$

In order to update the pheromone score, the following equation is used [9][10].

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \quad \text{....(7)}$$

Where $\Delta\tau_{ij}^h = \frac{1}{J^h}$, J^h is an objective function score, score $\alpha \ge 0$ and score $\beta > 0$.

In Ant Colony Optimization Clustering (ACOC) algorithm, the solution space is modeled as a graph with an object-cluster node matrix. The number of rows is similar to *m*, while the number of columns is similar to *g*. Each node is represented by *N(i,j)* meaning that Data Object *i* is determined at Cluster *j*. Each ant only occupies one of Nodes *g* of each object. In Figure 1, the graph construction of clustering problems is illustrated. It is noted that blank circles indicate not visited nodes, whereas bold circles indicate visited nodes by ants. Based on clustering results in Figure 1, a formed string solution is (2,1,2,1,3,3).

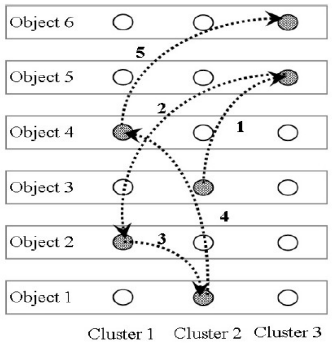

Fig. 1. Graph Construction of ACOC [7]

In the graph, every ant moves from one node to the others, leaves the pheromone at nodes, and forms a solution for the next step. In each step, it selects an object with no group at random and adds a new node to some solutions based on pheromone and heuristic intensity information.

In ACOC, ants leave the pheromone of nodes. Those with high pheromone are more attractive to ants. ACOC applies a pheromone matrix storing pheromone scores. Heuristic information indicates the desire to determine the data object at a certain cluster. Thus, calculation of Euclidean distance is among clustered data object and each cluster centroid of some ants. Nodes with higher heuristic scores are selected by ants. Each ant will bring a cluster centroid matrix (C^k) to store cluster centroids and change scores of clustering steps.

A complete procedure of ACOC is described as follows:

Step 1 : Conduct initialization of all ants. Make new iteration to find a number of ants. Initialize the pheromone matrix of each ant. Determine elements of the pheromone matrix with low scores (τ_0).

Step 2 : Conduct normalization of the pheromone matrix by using equation (3).

Step 3 : Initialize solution string to each ant at random. Compute the matrix weight (W^k) of each ant and the cluster centroid matrix (C^k) through equation (2) and, where *k=1..R*. *R* is the number of ants, $R \le m$.

Step 4 : Take Steps 2 and 3 until iteration reaches the number of ants.

Step 5 : Start new iteration. Compute the distance of data matrix and cluster centroid matrix through equation (4).

Step 6 : Compute selection of Cluster *j*. To determine *j* for selected *i*, there are two strategies used such as exploitation and exploration. Raise random number *q*. If $q < q_0$, exploitation is computed through equation (5). If not, exploration is computed through equation (6).

Step 7 : Form solution string from cluster selection. Create the weight matrix (W^k) of each ant. Fix the cluster centroid matrix (C^k).

Step 8 : Compute an objective function of each ant through equation (1). Next, have an ascending order of all objective function scores of all ants. The best solution string can be seen from the highest objective function scores.

Step 9 : Update the pheromone matrix through equation (7), where ρ is the pheromone evaporation rate in the range of 0 and 1 ($0.0 < \rho < 1.0$).

Step 10 : Take Steps 5 to 9. If all iterations are maximal, the clustering process stops. Then, take the solution string based on the best objective function.

ISODATA Algorithm

Forgy algorithm is one of simple clustering methods [11]. Besides using the data, *k*, the number of clusters formed becomes the input. Sample *k* is called as the seed point selected at random to support cluster selection.

ISODATA (Iterative Self-Organizing Data Analysis Techniques) algorithm is developed based on Forgy and K-means [11]. ISODATA algorithm minimizes the squared error.

Steps of Forgy algorithm are as follows [11]:

Step 1 : Initialize the cluster centroid to the seed point.

Step 2 : Find out the nearest cluster centroid for each data sample. Determine the data sample in the cluster based on the nearest cluster centroid.

Step 3 : If there is no change of cluster of the data sample, the process stops.

Step 4 : Compute the cluster centroid score based on clustering results. Return to Step 2.

Parameters used in ISODATA are as follows:

K = The number of clusters

I = The number of maximal iteration

P = The number of a maximal pair of data that can be combined

θ_N = The threshold score of the number of the minimal data sample in each cluster

θ_S = The threshold score of standard deviation (used for Split Operation)

θ_C = The threshold score of distance comparison (used for Merge Operation) .

Steps of ISODATA algorithm are as follows:

Step 1 : Initialize the number of clusters (*k*) and cluster centroids $m_1, m_2, ..., m_k$ to the seed point of the data sample ($x_i, i=1,2,...,N$).

Step 2 : Find out the nearest cluster centroid for each data sample. Determine the data sample in the cluster based on the nearest cluster centroid.

$x \in \omega_j$, if $D_L(x,m_j) = \max\{D_L(x,m_i)$, i=1,...,k

Step 3 : If the members in the cluster are less than θ_N, the cluster is ignored.

If each *j*, $N_j < \theta_N$, ignore ω_j and $k \leftarrow k-1$.

Step 4 : Compute cluster centroid scores based on clustering results through equation (8).

$$m_j = \frac{1}{N_j}\sum_{x\in\omega_j} x, \ (j = 1,...,k) \quad \text{..... (8)}$$

Step 5 : Compute an average distance of each data sample of the cluster with the cluster centroid through the equation (9).

$$D_j = \frac{1}{N_j}\sum_{x\in\omega_j} D_L(x,m_j), \ j=1,...,k \quad \text{..... (9)}$$

Step 6 : Compute an overall distance of the data sample and the representative of the cluster centroid through the equation (10).

$$D = \frac{1}{N}\sum_{j=1}^{k} N_j D_j \quad \text{..... (10)}$$

Step 7 : If there are only few clusters, continue with Step 8. However, if there are too many clusters, continue with Step 11. If not, continue with Step 14. Steps 8 to 10 are for Split operation, while Steps 11 to 13 are for Merge operation.

Step 8 : This is the first step of conducting Split operation. Determine the standard deviation vector $\sigma_j = [\sigma_1^{(j)},...,\sigma_n^{(j)}]^T$ for each cluster through equation (11).

$$\sigma_i^{(j)} = \sqrt{\frac{1}{N_j}\sum_{x\in\omega_j}(x_i - m_i^{(j)})^2}, \ i=1,...,n; \ j=1,...,k \quad (11)$$

Where $m_i^{(j)}$ is the –*i* component from m_j and σ_i is the standard deviation of the sample in ω_j as long as it is at the –*i* axis

coordinate. N_j is the number of the data sample in ω_j.

Step 9 : Find out a maximal component of each $\sum_j$ determined with $\sigma_{\max}^{(j)}$. Conduct this for all *j=1,...,k.*

Step 10 : If $\sigma_{\max}^{(j)} > \theta_S$, $D_j > D$, and $N_j > 2\theta_N$, split m_j into two new cluster centroids m_j^+ and m_j^- by adding $\pm\delta$ to Component m_j corresponding to $\sigma_{\max}^{(j)}$, where δ is in the form of $\alpha\sigma_{\max}^{(j)}$ and $\alpha > 0$. Next, delete m_j and $k \leftarrow k+1$. Return to Step 2. If not, continue with Step 14.

Step 11 : This is the first step of Merge operation. Compute the pair of distance D_{ij} between each of the two cluster centroids.
$D_{ij} = D_L(m_i, m_j)$, for each $i \neq j$, and arrange the distance $k(k-1)/2$ in an ascending format.

Step 12 : Find out the score that is less than the least P, D_{ij} and less than θ_C. Have an ascending order $D_{i_1 j_1} \leq D_{i_2 j_2} \leq ... \leq D_{i_P j_P}$.

Step 13 : Conduct pairwise merge operation: for l=1,..., P, with this step:
If m_{il} or if not m_{jl} is used in this iteration, combine them both into a centroid through the equation (12).

$$m = \frac{1}{N_{il} + N_{jl}} [N_{il} m_{il} + N_{jl} m_{jl}] \quad \text{.....(12)}$$

Delete m_{il}, m_{jl}, and $k \leftarrow k-1$. Return to Step 2.

Step 14 : The process is finished if the maximal Iteration I is reached. If not reached, return to Step 2.

III. RESULTS AND DISCUSSION

Created software in this research was a searching application used to collect library documents. In designing applications through this digital library, Apache Lucene was used to conduct indexing and searching. Next, in order to conduct clustering documentation, algorithms of Ant Colony Optimization, Forgy, and ISODATA were used. Created application architechture can be seen from Figure 2.

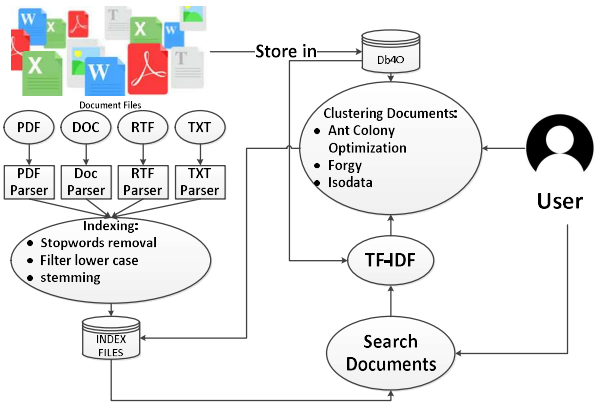

Fig. 2. Software Architecture of Document Clustering

There are two parts of this software architecture of document clustering such as Clustering and Searching. Documents including theses, thesis proposals, journal articles, and eBooks are firstly through text extraction (Parser) and analyzed through Lucene Library. Analysis steps include stopword process, standard analysis, lower case, and porter stemming. Results are further indexed and stored on index files. Searching process consists of Simple Search and Advanced Search. Simple Search uses Db4o conduct searching, while Advanced Search uses searching of terms in indices applying Lucene Library. Moreover, clustering process uses the matrix of index term frequency as the data matrix.

Simple Search is used to search documents based on keywords of titles or writers. Implementation of Simple Search covers SODA Object Query API from Object Database Db4o. The query process through SODA is quicker because it is conducted at API Levels. Meanwhile, Advanced Search is used to search documents based on keywords of content. In order to conduct the search, Query of Lucene is used. Query of Lucene provides simple features to determine the search of a certain field. The default field of Lucene is content. Search results are temporarily stored by using Hit List of Lucene.

Clustering Process used in this research is shown in Figure 3.

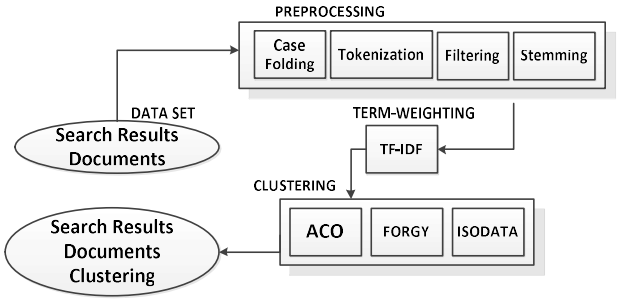

Fig. 3. Clustering Process

At a preprocessing stage, data are firstly processed. There are four steps of preprocessing such as case folding, tokenization, filtering, and stemming. Preprocessing produces outputs of bag-of-word such as the matrix containing words processed in this research. After bag-of-word is obtained, term weighting is conducted. At this stage, the weight of words is measured through TF-IDF. This results in term-weight-matrix,

namely the matrix consisting of the weight of words on documents. Next, term-weight-matrix is processed through one of clustering methods such as algorithms of Ant Colony Optimization, Forgy, and ISODATA. Results of clustering process are cluster lists of processed data.

The system was examined by using document clustering based on the case study of documents. This examination was conducted to prove system performance of applications in the implementation stage. Examination was conducted through observation of formed clusters. Therefore, the relationship of each document of each cluster could be seen and compared.

For the clustering examination, Ant Colony Optimization algorithm with the number of clusters (K) = 5, the number of iterations = 5, $\alpha = 1$, $\beta = 2$, the number of ants = 5, $\rho = 0.1$, and $q_0 = 0.01$ was in use.

Results of document clustering examination were obtained from Ant Colony Optimization algorithm in terms of processing time, the variance determining ideal clusters, and the sum of squared errors. The results can be seen from Table I.

TABLE I
CLUSTERING EXAMINATION OF ANT COLONY OPTIMIZATION

No	Number of Documents	Time	Variance	SSE
1	50	1m:42s	7.621419E-5	200,49818
2	100	5m:58s	5.0215316E-5	322,69882
3	200	9m:16s	1.4765678E-4	675,2462
4	500	22m:22s	1.3339706E-4	1147,4847
5	750	34m:31s	4.945447E-4	529,68787

Notes: m = minute; s = second

Document clustering examination using Forgy algorithm with the number of clusters (K) = 5 and the maximal number of iterations = 5 also yielded results in terms of processing time, the variance determining ideal clusters, and the sum of squared errors. They are shown in Table II.

TABLE II
CLUSTERING EXAMINATION OF FORGY ALGORITHM

No	Number of Documents	Time	Variance	SSE
1	50	0m:10s	6.074375E-5	155,71596
2	100	0m:35s	4.600567E-5	206,0978
3	200	1m:8s	1.2719215E-4	405,75742
4	500	2m:44s	1.18408505E-4	972,01855
5	750	4m:9s	4.91957E-4	527,5206

Notes: m = minute; s = second

Clustering examination through ISODATA algorithm applied number of clusters (K) = 5, the maximal number of iterations = 5, the minimal threshold numbers = 2, standard deviation threshold = 1, minimal distance threshold = 0.5, and the maximal number of threshold = 1.

Alike the previous algorithms, results of document clustering examination through ISODATA algorithm were processing time, the variance determining ideal clusters, and the sum of squared errors. The results can be seen from Table 4.

TABLE III
CLUSTERING EXAMINATION OF ISODATA ALGORITHM

No	Number of Documents	Time	Variance	SSE
1	50	0m:10s	6.181103E-5	130,9627
2	100	0m:41s	1.19376695E-4	203,49352
3	200	1m:15s	1.2797963E-4	406,49414
4	500	3m:32s	1.20486904E-4	959,76276

Notes: m = minute; s = second

Data provided in Tables I, II and III were summarized in the graphic of clustering duration. Figure 4 shows the comparison graphic of the three clustering methods. It is clear that Ant Colony Optimization algorithm tends to require longer time exponentially along with more increasing number of documents in comparison to other methods. On the other hand, ISODATA and Forgy methods tend to linearly improve as the number of documents increase.

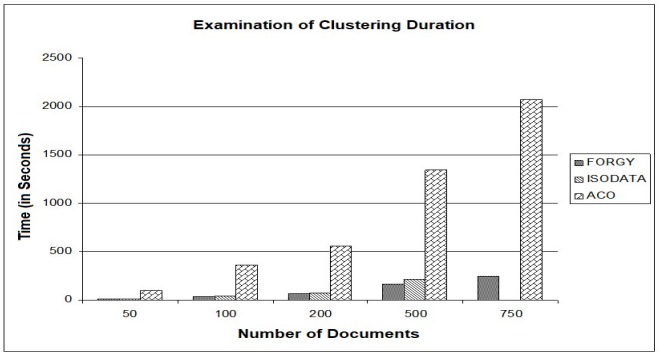

Fig. 4. Comparison Graphic of Clustering Duration

Based on clustering results of documents in Tables 1, II, and III, the comparison graphic of examination of variance determining ideal clusters of the three clustering algorithms could be presented. Figure 5 shows the comparison graphic of ideal variance based on clustering results of documents. It is obvious that the three algorithms tend to have almost similar variance.

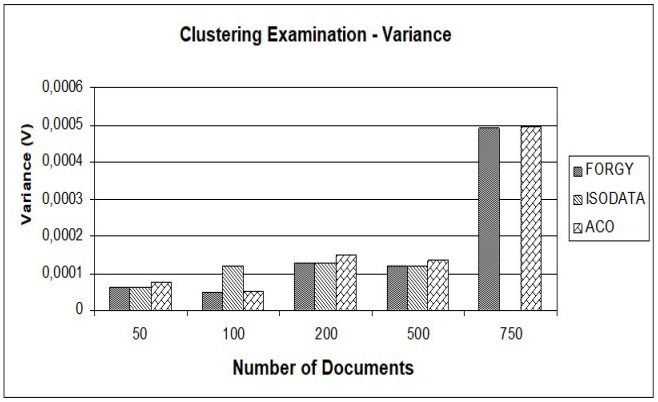

Fig. 5. Comparison Graphic of Variance

Based on clustering results of documents in Tables I, II, and III, the comparison graphic of examination of SSE determining ideal clusters of the three clustering algorithms could be presented. Figure 6 shows the comparison graphic of SSE based on clustering results of documents. It is obvious that Ant Colony Optimization algorithm tends to have higher SSE in comparison to others. However, ISODATA and Forgy algorithms have almost similar SSE. Clustering results are fine if SSE is lower.

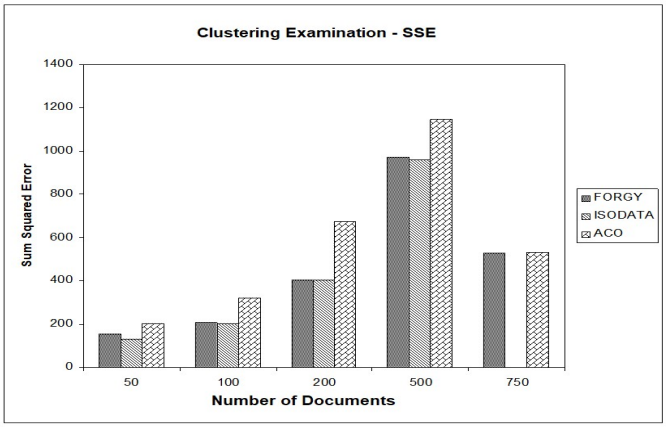

Fig. 6. Comparison Graphic of SSE

Research on document clustering with combination of algorithms of Ant Colony Optimization, Forgy, and ISODATA shows relationships of documents. Clustering results indicate existence of similar documents representing similarity of documents. However, the use of all words of searched documents is less accurate. This can result in documents with different content in each cluster because of insignificant words processed.

IV. CONCLUSION

Based on the research results and examination on clustering system of documents through Ant Colony Optimization, Forgy and ISODATA algorithms, it can be concluded that:

a. Clustering system through Ant Colony Optimization and ISODATA algorithms produces uncertain clusters, while Forgy algorithm always produces fixed clusters.
b. Clustering through Ant Colony Optimization algorithm requires very long processing time because of numerous ant colonies and iterations.
c. Generally, clustering results of the three methods possess almost similar variance in which each produces high intraclass similarity and low interclass similarity. The basis is the produced distance between data vectors and cluster centroids.
d. Clustering results through Ant Colony Optimization algorithm are better than ISODATA and Forgy algorithms. The indicator is the sum of squared errors of a number of documents and datasets examined.
e. To improve more relevant clustering results of documents, modification of clustering algorithms should be conducted.

REFERENCES

[1]. A. Singhal, et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 2001, 24.4: 35-43.
[2]. P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni. "An ant colony approach for clustering". *Analytica Chimica Acta*, 2004, 509.2: 187-195.
[3]. T.A. Runkler. "Ant colony optimization of clustering models". *International Journal of Intelligent Systems*, 2005, 20.12: 1233-1251.
[4]. P.S. Bradley, U.M. Fayyad. Refining Initial Points for K-Means Clustering. In: ICML. 1998. p. 91-99.
[5]. N. Memarsadeghi, D.M. Mount. A fast implementation of the ISODATA clustering algorithm. International Journal of Computational Geometry & Applications, 2007, 17.01: 71-103.
[6]. J. Ramos. "Using TF-IDF to determine word relevance in document queries." *Proceedings of the first instructional conference on machine learning*. Vol. 242. 2003.
[7]. Y. Kao., K. Cheng. "An ACO-based clustering algorithm." *International Workshop on Ant Colony Optimization and Swarm Intelligence*. Springer, Berlin, Heidelberg, 2006.
[8]. T. Stutzle. "Ant colony optimization". In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, Berlin, Heidelberg, 2009. p. 2-2.
[9]. E. Bonabeau., M. Dorigo, G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. No. 1. Oxford university press, 1999.
[10]. M. Dorigo, and L.M. Gambardella. "Ant colony system: a cooperative learning approach to the traveling salesman problem." IEEE Transactions on evolutionary computation 1.1 (1997): 53-66.
[11]. G. Earl., R. Johnsonbaugh., S. Jost. "*Pattern recognition and image analysis*." Prentice Hall (1996).
[12]. G.H. Ball., D.J. Hall., 1965, "ISODATA: A novel methods of data analysis and pattern classification", *Technical Report AD0699616*, Stanford Research Institute, Stanford, CA, U.S., April 1965.
[13]. J. Han, , J. Pei, M. Kamber. *Data mining: concepts and techniques.* Elsevier, 2011.
[14]. G. Kekec., N. Yumusak, N. Celebi. "Data Mining and Clustering With Ant Colony." *Proceedings of 5th International Symposium on Intelligent Manufacturing Systems*. 2006.
[15]. N. Memarsadeghi, D.M. Mount., N.S. Netanyahu., J. Le Moigne. "A fast implementation of the ISODATA clustering algorithm." *International Journal of Computational Geometry & Applications* 17.01 (2007): 71-103.
[16]. W.H. Ming., C.J. Hou. "Cluster analysis and visualization." *Workshop on Statistics and Machine Learning, Institute of Statistical Science, Academia Sinica*. Vol. 20041. 2004.
[17]. N. Monmarché. "On data clustering with artificial ants." *AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms*: Research Directions. 1999.
[18]. A. Passadore., G. Pezzuto. "An indexing and clustering architecture to support document retrieval in the maintenance sector." *FSESUPPORT at Maintenance Management* 2007 Event (2007).
[19]. A.R.D. Prasad, D. Patel. "Lucene search engine: An overview." *DRTC-HP International Workshop on Building Digital Libraries using DSpace* (2005). 7th – 11th March, 2005, DRTC, Bangalore.
[20]. C.F. Tsai., H.C. Wu, C.W. Tsai. "A new data clustering approach for data mining in large databases." *Parallel Architectures, Algorithms and Networks, 2002. I-SPAN'02. Proceedings. International Symposium* on. IEEE, 2002.