Asian Journal of Information Technology 13 (1): 21-28, 2014
ISSN: 1682-3915

Text Document Clustering with Flocking Algorithm using Specific Crimes Judgment Corpus

[1]S. Vijayalakshmi and [2]D. Manimegalai
[1]Department of Applied Sciences, Sethu Institute of Technology, Kariapatty, India
[2]Department of Information Technology, National Engineering College, Kovilpatty, India

Abstract: Text document clustering is the fundamental technique to mine massive amount of textual data. The problem is of high dimension and most of the machine learning algorithms does not perform well with all the terms in the corpus. In this study, researchers proposed an application of flocking algorithm for text document clustering using two document representation methods. They are Unigram and Noun. In this research, the problem of high dimensions has been dealt with representing documents as Bag of Nouns (BoN) and Bag of Unigrams (BoU). As there are thousands of words present in documents to find Unigram, user has to connect with WordNet and verified the selected features are Unigram. The same process is repeated for Noun. In clustering algorithm, boids follow four simple local rules like alignment, separation, cohesion and similarity to calculate the velocity for flocking. Experiments were conducted with documents of 20 Newsgroup, Reuter Real datasets and Specific Crime Judgment corpus to study the advantages of the system. Flocking algorithm for Text Document clustering is compared with Unigram based document representation and Noun based Document representation. It is observed that Flocking algorithm with Bag of Noun is working efficiently than Bag of Unigram and Bag of Words.

Key words: Clustering, Flocking algorithm, RiTa WordNet, HCLK-mean, K-mean

INTRODUCTION

How to explore and utilize the huge amount of text documents is a major question in the area of information retrieval and text mining. Document clustering (Konchady, 2007) is one of the most important text mining methods. Clustering is mainly to group all objects into several mutually exclusive clusters in order to achieve the maximum or minimum of an objective function. Flocking is the phenomenon in which self-propelled individuals using only limited environmental information and simple rules, organize into an ordered motion. Flock (Davison, 2005) is a collective noun for various animal groups of birds, group of fish, a herd of sheep, goats or similar animals, a crowd of people. In this implementation, each document is a boid (bird-oids) and flies toward other document that are similar to it.

Text clustering (Agarwal and Yu, 2000) has attracted more and more attention from researchers due to its wide applicability. It widely used in the machine learning community have been applied such as Naive Bayes based clustering, Neural Network based clustering, k Nearest Neighbour based clustering and traditional clustering such as Partitioning and Hierarchical. Recently, Cui *et al.* (2006) and Erra *et al.* (2011) have been obtained excellent results by using Flocking algorithm based clustering. While wide range of clustering has been used virtually all of them were based on the same text representation called Bag of Words where set of words appearing in the document. The contribution this study are the following:

- Unigram features and Noun features were designed to represent document and unsupervised text document clustering using Flocking algorithm are designed which supports automatic generation of clusters and also designed to generate expected number of cluster with prior knowledge given by user. These features can help to improve the performance
- Synthetic dataset is designed and called as Specific Crime Judgment Corpus (SCJC). These judgments are published by all High Courts in India. Researchers have taken main five classes: offence against state, offence against property, offence against human body, offence relating to religion, offence relating to marriage. This judgment may useful for advocates and Judges for their reference

Corresponding Author: S. Vijayalakshmi, Department of Applied Sciences, Sethu Institute of Technology, Kariapatty, India

- The factor affecting the performance of Unigram features and Noun features are discussed. The benefit of these features is closely related to the length of the document in the corpus and the writing style of the document

Reynolds (1987) presented a model of polarized, non-colliding aggregate motion such as flocks, herds and schools. The model (Reynolds, 1987; Motsch and Tadmor, 2011) is based on simulating the behaviour of each bird independently and working independently. The birds try both to stick together and avoid collisions with one another and with other objects in their environment. The animations showing simulated flocks built from this model seem to correspond to the observer's intuitive notion of what constitutes flock-like motion. However, it is difficult to objectively measure how valid these simulations are by comparing behavioural aspects of the simulated flock with those of natural flocks. Researchers are able improve and refine the model. But having approached a certain level of realism in the model. The parameters of the simulated flock can be altered by the animator to achieve many variations on flock-like behaviour.

Cui *et al.* (2006) presented flocking based clustering. In this algorithm, each document represented as boids. Each boids follow four simple local rules. They are alignment, separation, cohesion and the feature similarity and dissimilarity rule to move in the virtual space and also form a flock or cluster.

Erra *et al.* (2011) proposed a nature inspired clustering model (Erra *et al.*, 2011) and presented an efficient implementation for the GPU. Each document considered as agent. Agent following the above specified simple rules emerge a complex global behaviour and agents similar to each other gradually merge together to form a cluster. GPU is based on static grid that tackles the problem of the identifying neighbour.

Amiri *et al.* (2009) proposed an application of Shuffled Frog-Leaping Algorithm (SFLA) in clustering. The SFLA (Amiri *et al.*, 2009) has been designed as a meta-heuristic to perform an informed heuristic search using a heuristic function (any mathematical function) to seek a solution of a combinatorial optimization problem. It is based on evolution of memes carried by the interactive individuals and a global exchange of information among themselves. Shuffled frog-leaping algorithm is considered as a typical swarm-based approach to optimization. The shuffled frog-leaping algorithm draws its formulation from two other search techniques: the local search of particle swarm optimization and the competitiveness mixing of the shuffled complex evolution technique. The SFL algorithm for data clustering can be applied when the number of clusters is known a priori and are crisp in nature.

MATERIALS AND METHODS

Text document clustering with document representation using Noun and Unigram: Researchers used RiTa WordNet ontology (Howe, 2009) to explore the Noun and Unigrams.

Overview of text clustering: In most existing Text Clustering algorithm, text documents are represented by using vector space model and each document d is considered as a vector in the term space. In this model, researchers construct two types of document representation model instead of terms, Noun and Unigram extracted by RiTa WordNet. Here, each document is a vector in the Noun space and Unigram space. It is represented by noun frequency (nf) vector and unigram frequency (uf) vector, respectively:

$$d_{nf} = [nf_1, nf_2, nf_3,..., nf_n] \quad (1)$$

$$d_{uf} = [uf_1, uf_2, nu_3,..., uf_n] \quad (2)$$

Where:

nf_i, uf_i = The frequency of the ith term in the document
n = The dimension of the text database which is the total number of unique terms

This model follows the pre-processing steps: including tokenize, removal of stopword, extract Nouns and extract Unigrams by using RiTa WordNet on the documents. A widely used refinement model (Welling, 2010) is applied to weight each Noun based/Unigram based on its Inverse Document Frequency (IDF) in the corpus. To account the document of different lengths, the length of the each document vector is normalized to a unit length. In the rest of study, researchers assume this normalized vector space model weighted by NF-IDF/UF-IDF is used to represent documents during clustering. The most commonly used is the cosine function to measure the similarity between the documents and the correlation between the document vectors representing them. The similarity between the two documents d_i and d_j are calculated as:

$$\text{cosine}(d_i, d_j) = \frac{d_i * d_j}{\|d_i\|\|d_j\|} \quad (3)$$

where, * represents vector dot product and $\|d_i\|$ denotes the length of the vector d_i. The cosine value is 1 when two documents are identical and 0 is nothing in common between them. The larger cosine value indicates that these two documents share more terms and are more similar.

The K-Means algorithm is very popular for solving the problem of clustering a dataset into k clusters, if the dataset contains n documents d_1, d_2, d_3, ..., d_n then the clustering is the optimization process of grouping them into k clusters so that the global criterian function:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} f(d_i,\ cen_j) \tag{4}$$

is either minimized or maximized. cen_j represents the centroid of the cluster c_j for j = 1, 2, ..., k and $f(d_i, cen_j)$ is the clustering criterion function for the document d_i and a centroid cen_j. When the cosine function is used each document is assigned to the cluster with the most similar centroid and the global criterion function is maximized as a result. The optimization process is called NP-Complete problem.

HCLK-Mean algorithm: Competitive learning (Zhong, 2005) is a rule based on the idea that only one neuron from a given iteration in a given layer will fire at a time. There are weights which are adjusted such that only one neuron in a layer for instance the output layer, fire. In terms of neural computing, there are several different and often mutually exclusive goals which can be set for competitive learning systems such as error minimization, entropy maximization, feature mapping and other goals. Also there are some different learning methods such as hard competitive learning, soft competitive learning without fixed dimensionality and soft competitive learning with fixed dimensionality. Winner take all algorithm was introduced by Kohonen, Hecht-Nielsen and is an unsupervised learning. It also called as hard competitive learning and commonly used in computational models of the brain, particularly for distributed decision-making in the cortex. It has been formally proven that the winner take all operation is computationally powerful compared to other nonlinear operations such as thresholding.

It receives set of N unit-length the data vectors for X = $\{x_1, x_2, ..., x_n\}$ as input. The vector x is designed by using only noun phrase which must be available in RiTa WordNet. The HCLK-Mean algorithm aims to maximize the average cosine similarity objective y_n = arg $\max_k x_n^T\mu_k$. This model incrementally updates the closest cluster center μy_n as:

$$\mu_{yn} = \frac{\mu_{yn}+\alpha x}{\left\|\mu_{yn}+\alpha x\right\|}$$

for given data point x. In many cases, calculating distances rather than comparing activation levels on normalization weight vectors is more efficient to determine the winner. The α is a small positive learning rate that usually decreases as the learning proceeds. The process of decreasing the learning rate over time is called annealing the learning rate and it necessary for learning to reach its optimal solution. The decreased rate of learning:

$$\alpha^{(t)} = \alpha_s\left(\frac{\alpha_f}{\alpha_s}\right)$$

Where:

t = The iteration index must be between s and f
s = 0 = The starting iteration value
f = The final iteration value

This learning rate is work better than flat learning rate.

Flocking algorithm: Each entity (document) is called as a boid, moves around while being governed by a few simple rules. Boids is an artificial life simulation originally developed by Reynolds (1999). The aim of the simulation was to replicate the behaviour of flocks of birds. Flocks (Cui *et al*., 2006) of birds can be modelled with boids. The boids simulation only specifies the behaviour of each individual bird. With only a few simple rules, the program manages to generate a result that is complex and realistic enough to be used as a framework for computer graphics applications such as computer generated behavioural animation in motion picture films. Each boid starts out at the centre of the map with a random velocity and for each frame of the simulation, a new velocity is calculated using the flocking algorithm. For each boid, the algorithm uses the boid's current velocity, its neighbours' velocities and its position relative to its neighbours to calculate this new velocity by using three simple rules called alignment, cohesion and separation when properly applied, produce realistic-looking flocking behaviour. The boids program consists of a group of objects (birds) that each has their own position, velocity and orientation. In existing algorithm, the boid that share similar document vector feature will automatically group together into a cluster by using the above specified three rule with traditional document representation with TF-IDF style.

The other boids with different document vector features will keep away from this flock. The clustering approach, the document NF-IDF/UF-IDF vector is represented as the feature of the boid. The proposed algorithm includes four rules and also implements a simple local propagation algorithm for cluster identification and generates the cluster automatically. The resultant number of cluster may differ from initial partition of the dataset. If the user wants to get expected number of cluster, prior knowledge about how many clusters are expected in the dataset and the initial partition of the dataset.

Each boid has strictly local perception of the space it occupies. The behaviour of each boid B with position pos_b is influenced by all boid X with in a pos_x in its neighbourhood. It is driven by a set of local behaviour rule. Each of the boids rules works independently. Researchers can calculate how much it will get moved by each of the four rules, giving us four velocity vectors for each boid then researchers can add those the four vectors to the boid's current velocity to research out its new velocity.

Attraction: Every boid attempts to move towards the average position of other nearby boids toward boids within a short radius. These rules follow the constraints that each boid has a maximum velocity and acceleration. The centre of mass is simply the average position of all the boids. Lets researchers have N boids called b_1, b_2, b_3, ..., b_N. Also, the position of a boid b is denoted b.pos. Then, the centre of mass c of all N boids is given by:

$$v = \frac{(b1.pos + b2.pos + ... + bN.pos)}{N} \tag{5}$$

Remember that the positions here are vectors and N is a scalar. However, the centre of mass is a property of the entire flock. This model would prefer to move the boid toward its perceived centre which is the centre of all the other boids not including itself. Thus, $boid_j$ $(1 \leq J \leq N)$, the perceived centre v1 is given by:

$$v1 = \frac{b_1.pos + b_2.pos + ... + b_{J-1}.pos + ... + b_N.pos}{N} \tag{6}$$

For example, researchers need to research out how to move the boid towards it to the perceived centre, move it 1% of the way towards the centre this is given by:

$$\frac{(v1 - b_J.position)}{100}$$

The mathematical implementation is:

$$d(b_x, b_j) \leq d1 \cap d(b, b_j) \geq d1 \rightarrow v1 = \frac{1}{n}\sum\nolimits_{x=1}^{n} V_x \tag{7}$$

Collision avoidance between boids: Each boid attempts to maintain a reasonable amount of distance (example 100) between itself and any nearby boids to prevent overcrowding, this process called as separation. If two boids within a defined small distance of another boid, move it as far away again as it already is. This is done by subtracting from a vector v2 the displacement of each boid which is nearby. Researchers initialize v2 to zero as researchers want this rule to give us a vector which when added to the current position moves a boid away from those near it. The mathematical implementation is:

$$\text{Position distance } (b_x, b_j) = \sum_{x=1}^{n} \sqrt{(b_x.pos - b_j.pos)^2} \tag{8}$$

$$\text{If } (d(b_x,\ b_j) \leq d2 \rightarrow v2 = v2 - \text{position distance } (b_x, b_j) \tag{9}$$

If two boids are near each other, this rule will be applied to both of them. Hence, the resultant repulsion takes the form of a smooth acceleration. It is a good idea to maintain a principle of ensuring smooth motion. Instead, researchers have them slow down and accelerate away from each other until they are far enough apart for the penchant.

Velocity matching: Boids try to change their position so that it corresponds with the average alignment of other nearby boids. It is also called as cohesion. The cohesion force is obtained by computing the average positions of neighbours. Researchers calculate a perceived velocity, v3 then add a small portion (Eq. 8) to the boid's current velocity:

$$d(b_x,\ b_j) \leq d1 \cap d(b_x,\ b_j) \geq d1 \rightarrow v3 = \sum_{x=1}^{n}(b_x - b_j) \tag{10}$$

The steering force SF of the Reynold Model for boid I is achieved by the sum of the steering forces produced by behaviors:

$$\text{Velocity} = v1 + v2 + v3 \tag{11}$$

Feature similarity: The boid velocity is also impacted by the feature similarity compared to the nearby boids. The boids tries to stay close toother boids that have similar features. For Document Clustering algorithm, the boid's feature is represented by a document NF-IDF/UF-IDF vector. The similarity between two boids is computed by using values of the associated feature vectors with the help of Eq. 3.

Cluster identification: Researchers use a simple local propagation algorithm for cluster identification. The algorithm is composed by two steps:

- Assign a unique label into each boid
- Each boid looks to each of its neighbours in turn. If its neighbor's label is smaller than its own label then it replaces its label with that of its label with that of its neighbor, repeat this step at L times

In existing system, Flocking algorithm generate cluster without prior knowledge. The proposed algorithm can generate clusters automatically. But if the user wants to give expected number of cluster prior, user can assign it. Cluster identification will repeats until reach the expected number of cluster using the above algorithm.

Algorithm for flocking algorithm:

```
Input : D: Documents or part of documentd
        RD : RiTa WordNet Dictionary used
Output: LU (List of Unigram in document) based clusters as user specified prior
//Same process will be repeated with noun called LN based clustering with flocking.
For each word $W_i$ in D
If $W_i$ is not stopword then
Determine its POS using RiTa WordNet
If $W_i$ is Noun |verb| adjective| adverb then
Stem word of $W_i$ and store in $LU_D$ //In LN based clustering to be
    considered only Noun, stem and store in $LN_D$
Construct Noun Document representation and Unigram Document
    representation using the extracted features
Calculate weight using tf-idf style.
Start flocking
Initialise positions of boids
Loop starts
Vector v1, v2, v3, v4;
Boid b
Construct Boids and Repeat for each Boid b
v1 = rule1 (b);
v2 = rule2 (b);
v3 = rule3 (b);
b. velocity = b. velocity+v1+v2+v3;
v4 = Calculate the similarity using cosine formula
Identify cluster based on agglomerative style using Local Probagation
    algorithm
    //To get expected number of clusters
b. velocity = b. velocity+v1+v2+v3+v4;
limit_velocity (b)
b. pos = b. pos+b. velocity
End Loop
End of Algorithm
```

Limiting the speed: The speed of the flocking will actually fluctuate and it is possible for them to momentarily go very fast. The real animals can not go arbitrarily fast actually and so researchers limit the boids' speed. The definitions of velocity are a vector and thus have both magnitude and direction whereas speed is a scalar and is equal to the magnitude of the velocity:

$$\text{If } |\text{b.velocity}| > \text{v lim then b.velocity} = \left(\frac{\text{b.velocity}}{\text{b.velocity}}\right) * \text{v lim} \tag{12}$$

This algorithm creates a unit vector by dividing b. velocity by its magnitude and then multiplies this unit vector by vlim. The resulting velocity vector has the same direction as the original velocity but with magnitude vlim. This limiting speed algorithm is called after all the other rules have been applied and before calculating the new position:

$$\text{b.velocity} = \text{b.velocity} + \text{v1} + \text{v2} + \text{v3} \tag{13}$$

RESULTS AND DISCUSSION

Performance measure: Each cluster (Li *et al*., 2008) obtained can be considered as a result of a query whereas each pre-classified set of documents can be considered as the desired set of documents for that query. Thus, researchers can calculate precision P(i, j) and recall R(i, j) of each cluster j for each class i. If n_i is the number of the member of the class i, n_j is the number of the number of the cluster j and n_{ij} is the number of the member of the class i in the cluster j.

F-measure:

$$P(i, j) = \frac{n_{ij}}{n_j} \tag{14}$$

where, n_j is the total number of documents obtained in cluster j:

$$R(i, j) = \frac{n_{ij}}{n_i} \tag{15}$$

where, n_i is the total number of documents obtained in class i:

$$F(i, j) = \frac{2 * P(i, j) * R(i, j)}{P(i, j) * R(i, j)} \tag{16}$$

F-measure for the whole clustering result is defined as:

$$\text{Overall F-measure} = \sum_i \frac{n_i}{n} \max\left(F(i, j)\right)$$

Purity:

$$\text{Purity}(j) = \frac{1}{n_j} \max(n_{ij}) \tag{17}$$

Overall purity (Manning *et al*., 2008) of clustering result is a weighted sum of purity values of cluster thus the purity of the clusterj is defined as:

$$\text{Overall purity} = \sum_{j} \frac{n_j}{n} * \text{Purity}(j) \qquad (18)$$

Entropy: Entropy of cluster j is calculated as:

$$E_j = -\sum_{j} \text{Precision}_{ij} * \log(\text{Precision}_{ij}) \qquad (19)$$

Overall entropy for a set of cluster can be defined:

$$\text{Overall entrophy} = \sum_{j} \frac{n_i}{n} * E_j \qquad (20)$$

Experimental setup: The research was implemented using Net Beans 7.2.1, 4 GB RAM in windows7 (64 bit) operating system with i5 processor. RiTa WordNet 3.0 is used to check noun, verb, adverb, etc. The text clustering approach used in this study has been implemented and evaluated with extensive experimentation using three of the following datasets. Two real dataset are 20 news group and Reuter-21578 and one synthetic dataset is Specific Crime Judgment Corpus (SCJC). These datasets are split based on its size like large datasets and smaller datasets.

In real datasets, this work uses two sub dataset of different nature which were formed from the popular Benchmark 20 Newsgroup. The first dataset is called Dataset1 which consist of six categories: rec.sport.hockey, rec.autos, sci.crypt, alt.atheism, comp.graphics and talk.politics.mideast of 20 newsgroup. The second dataset is called Dataset2 and consist of categories: sci.crypt, comp.graphics, comp.os, ms.windows.misc, comp.windows.x, comp.sys.ibm.pc. hardware and comp.sys.mac.hardware. It may be observed that the topics of the Dataset1 are different while five topics out of six. Dataset2 are related to computer science. Document from the two categories sci.crypt and comp.graphics were used in both datasets. Third dataset was formed with 10 categories from Reuters-21578. A subset of Reuters-21578 is currently the most widely used test collection for text categorization and clustering research. The data was originally collected and labelled by Carnegie Group, Inc. and Reuters. Because the dataset contains some noise such as repeated documents, unlabelled documents and nearly empty documents, researchers choose a subset of 10 relatively large groups (acq, coffee, crude, earn, interest, monet-fx, money-supply, ship, sugar and trade). The fourth synthetic dataset is Specific Crime Judgment Corpus (SCJC) provided by all High Courts in India (www.http://indiankanoon.org). The SCJC includes five categories: offence against state, offence against property, offence against human body, offence relating to religion, offence relating to marriage. Researchers used this SCJC that contains 100 each real judgments. These judgments are collected from the internet at different time stages and have been categorized by human experts and manually clustered into five categories. This corpus includes original judgment given by various High Courts in India. So, it may be useful for advocates and judges for their reference.

Fifth dataset is the subset of 7 relatively disjoint groups (comp.windows.x, rec.autos, sci.crypt, sci.med, talk.politics.guns, rec.sport.baseball and soc.religion. christian) each with exactly 100 documents. Researchers call this dataset NG700. The remaining two datasets are smaller in size. RDS all documents have at least 2 kB each. Last smallest dataset include hundred documents from SCJC, 20 documents from each category called tiny SCJC (Table 1).

The Flocking algorithm, HCLK-Means algorithm and K-Means algorithm are applied to the synthetic and real document collection, respectively. The cosine distance measure is used as similarity metric in each algorithm. Each document is represented as boid in Flocking algorithm. Each boid can only sense the flock mates located with sense range. Each boid may require may need more computational resources to calculate its flying direction and speed. In this implementation, researchers have used the boids ranges from 100-1200.

HCLK-Means clustering require learning rate as parameter. The learning rate is decreased during the process time to get improved optimal solution. All three algorithm required prior knowledge about how many clusters are expected in the dataset (Table 2, 3 and Fig. 1-4).

Table 1: Summary of large datasets used in experiments

Datasets	No. of doc. in dataset	Classes	Dataset size
SCJC (synthetic)	500	5	17.2 MB
Reuters (real)	1750	7	4.10 MB
NG700 (real)	700	7	3.2 MB
Dataset1 (real)	1200	6	2.73 MB
Dataset2 (real)	900	6	1.12 MB
Tiny SCJC (synthetic)	100	5	740 kB
RDS (real)	300	3	694 kB

Table 2: Overall F-measure, overall entropy and overall purity of TDC with Flocking algorithm

	Overall F-measure			Overall purity			Overall entropy		
Datasets	BoN	BoU	BoW	BoN	BoU	BoW	BoN	BoU	BoW
SCJC	0.34	0.31	0.32	0.35	0.33	0.32	0.64	0.66	0.94
Reuters	0.27	0.29	0.34	0.28	0.32	0.35	0.77	0.73	0.70
NG700	0.31	0.34	0.32	0.31	0.35	0.33	0.48	0.71	0.67
Dataset1	0.43	0.39	0.39	0.43	0.38	0.39	0.58	0.65	0.65
Dataset2	0.30	0.35	0.34	0.33	0.35	0.34	0.62	0.62	0.68
Tiny SCJC	0.40	0.35	0.35	0.40	0.40	0.36	0.59	0.61	0.87
RDS	0.43	0.38	0.33	0.67	0.40	0.37	0.44	0.47	0.47

Table 3: Comparisons of F-measure, purity and entropy

	F-measure			Purity			Entropy		
Datasets	FA	HCLK	K	FA	HCLK	K	FA	HCLK	K
SCJC	0.41	0.31	0.34	0.52	0.29	0.60	0.53	0.57	0.30
Reuters	0.37	0.23	0.25	0.40	0.31	0.21	0.68	0.79	0.80
NG700	0.39	0.29	0.28	0.50	0.29	0.26	0.44	0.34	0.74
Dataset1	0.63	0.29	0.29	0.94	0.27	1.00	0.12	0.47	0.52
Dataset2	0.40	0.27	0.28	0.58	0.28	0.45	0.45	0.45	0.59
Tiny SCJC	0.47	0.43	0.37	0.50	0.47	1.00	0.53	0.52	0.45
RDS	0.57	0.53	0.44	0.78	0.61	0.39	0.23	0.32	0.32

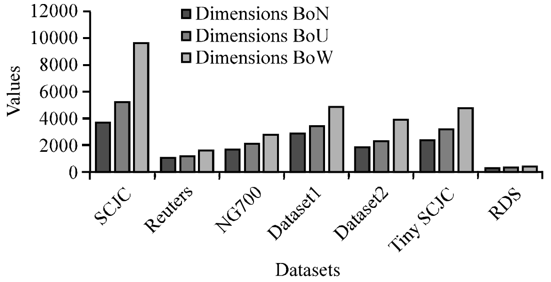

Fig. 1: Dimensions (or) unique feature used for TDC

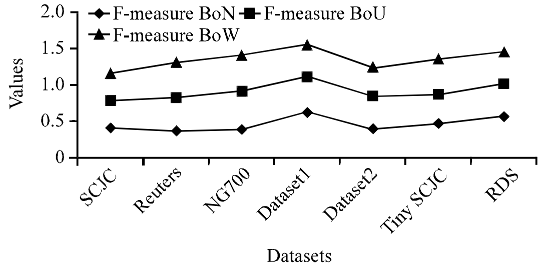

Fig. 2: F-measures of TDC with Flocking algorithm

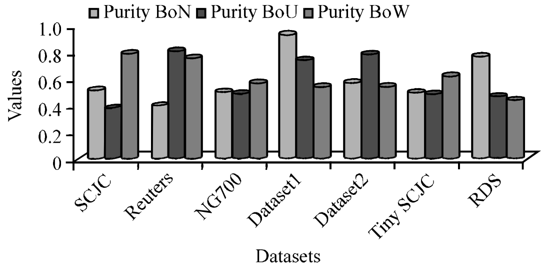

Fig. 3: Purity of TDC with Flocking algorithm

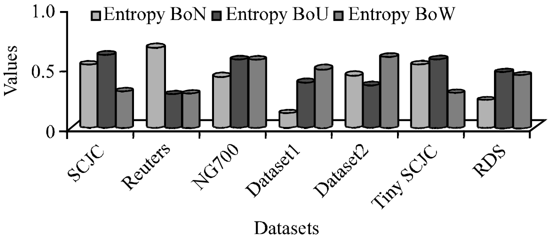

Fig. 4: Entropy of TDC with Flocking algorithm

CONCLUSION

Identifying optimal feature subset for text clustering in an NP-hard problem and becomes more difficult when the number of feature is less. In this study, Noun phrase feature representation with Flocking algorithm is proposed and its performance is better when compared with unigram based features with FA and also compared with the previous research Noun based features with HCLK-mean and K-mean. In future, analysis could be made to further improve clustering accuracy by using hybrid feature selection methods.

REFERENCES

Agarwal, C.C. and P.S. Yu, 2000. Finding generalized projected clustering in high dimensional space. Proceedings of the ACM SIGMOD International Conference on Management of Data, May 15-18, 2000, Dallas, TX., USA., pp: 70-81.

Amiri, B., M. Fathian and A. Maroosi, 2009. Application of shuffled frog-leaping algorithm on clustering. Int. J. Adv. Manuf. Technol., 45: 199-209.

Cui, X., J. Gao and T.E. Potok, 2006. A flocking based algorithm for document clustering analysis. J. Syst. Archit., 52: 505-515.

Davison, A., 2005. Java Gaming and Graphics Programming. In: Killer Game Programming in Java, Davison, A. (Ed.). O'Reilly Publication, USA.

Erra, U., B. Frola and V. Scarano, 2011. A GPU-based interactive bio-inspired visual clustering. Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, April 11-15, 2011, Paris, France, pp: 268-275.

Howe, D.C., 2009. RiTa: Creativity support for computational literature. Proceedings of the 7th ACM Conference on Creativity and Cognition, October 27-30, 2009, Berkeley, CA., USA., pp: 205-210.

Konchady, M., 2007. Text Mining Application Programming. Charles River Publisher, Boston, Massachusetts, ISBN-13: 9781584504603.

Li, Y., C. Luo and S.M. Chung, 2008. Text clustering with feature selection by using statistical data. IEEE Trans. Knowl. Data Eng., 20: 641-652.

Manning, C.D., P. Raghavan and H. Schutze, 2008. An Introduction to Information Retrieval. Cambridge University Press, USA., ISBN-13: 9780521865715, Pages: 482.

Motsch, S. and E. Tadmor, 2011. A new model for self-organized dynamics and its flocking behavior. J. Stat. Phys., 144: 923-947.

Reynolds, C.W., 1987. Flocks, herds and schools: A distributed behavioral model. Comput. Graph., 21: 25-34.

Reynolds, C.W., 1999. Steering behaviors for autonomous characters. Proceedings of the Conference on Game Developers, March 16-18, 1999, San Jose, California, pp: 763-782.

Welling, M., 2010. A first encounter with machine learning. Donald Bren School of Information and Computer Science, University of California, Irvine, USA.

Zhong, S., 2005. Efficient online spherical k-means clustering. Proceedings of the IEEE International Joint Conference on Neural Networks, Volume 5, July 31-August 4, 2005, Montreal, QC, Canada, pp: 3180-3185.