International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

# Document Class Identification using Fire-Fly Genetic Algorithm and Normalized Text Features

**Jitendra Yadav, Prof. Sumit Sharma, Prof. Pranjali Malviya**
Department of Computer Science & Engineering,
Vaishnavi Institute of Technology and Science (VITS), Bhopal, India.
Email Id: jtndr.gkg@gmail.com

*Abstract* - **Increase of digital platform resources ultimately raises the text content on different platform, so organization of this unstructured data is highly required. Researcher has proposed number of techniques to filter relevant data inform of feature so enhancing the understanding of work. This paper has work on clustering the document of research filed into respected class. Here paper has proposed a modified firefly genetic algorithm for clustering of document into respected cluster. Fire fly crossover operation was modified in this paper where best chromosome update rest of population. Pattern based features were evaluate from the text content. Normalized content feature were used for the work. Experiment was done on real set of research paper taken fro various research domains. Results were compared on different evaluation parameters with exsisting model and it was obtained that proposed model of FFDC (Fire Fly Document Classification) was better than other.**

*Keywords*- **Clustering, Genetic Algorithm, Text Mining, Pattern Feature.**

## I. INTRODUCTION

Unstructured data remains a challenge in almost all data intensive application fields such as business, universities, research institutions, government funding agencies, and technology intensive companies [1]. Eighty percent of data about an entity (person, place, or thing) are available only in unstructured form [2]. They are in the form of reports, email, views, news, etc. Text mining/ analytics analyzes the hitherto hidden relationships between entities in a dataset to derive meaningful patterns which reflect the knowledge contained in the dataset. This knowledge is utilized in decision making [3].

The amount of corporate data is constantly increasing, and the growing need for automation of complex data intensive applications drives the industry to look for better approaches for knowledge discovery. This knowledge will lead to insightful investments and increase the productivity of an organization. This article will also be helpful for researchers to understand the capacity of various text classification techniques before working with data intensive applications and their adaptability to AI procedures. The world requires more intelligent systems like "Siri"; therefore, developing AI-based text processing models are the need of the hour. Apart from these, the variety of data available, cheaper computational processing, and affordable data storage calls for automation of data models that can analyze complex data to deliver quick results. For this reason, this study analyzes text classification techniques with respect to AI/ML.

Text analytics converts text into numbers, and numbers in turn bring structure to the data and help to identify patterns. The more structured the data, the better the analysis, and eventually the better the decisions would be. It is also difficult to process every bit of data manually and classify them clearly. This led to the emergence of intelligent tools in text processing, in the field of natural language processing, to analyze lexical and linguistic patterns [4]. Clustering, classification, and categorization are major techniques followed in text analytics. It is the process of assigning, for example, a document to a particular class label (say "History") among other available class labels like "Education", "Medicine" and "Biology". Thus, text classification is a mandatory phase in knowledge discovery [5].The aim of this article is to analyze various text classification techniques employed in practice, their spread in various application domains, strengths, weaknesses, and current research trends to provide improved awareness regarding knowledge extraction possibilities.

## II. RELATED WORK

**In [6]** they used methods for classification of real-time data using convolution and recurrent neural networks. They explored, experimenting and providing new approaches of classification non-stationary data using the RNN neural network. Experimental result of F1 score in the case of CNN 0.8 and by LSTMs 0.92.

© 2020 IJSRET

**In [7]** the bidirectional recurrent neural networks (BRNN) are used to retrieve the past and future data while a convolutional layer is used to encapsulate local data. Here the standard RNN is replaced by two recently appeared RNN modifications, called long short-term memory (LSTM) and gated recurrent unit (GRU), to increase the effectiveness of the new architecture for real-time classification. The basic advantage is that the experimental model is trained end-to-end without human involvement and it is easily implemented.

**In [8]** system based on the ANTS algorithm and cluster mapping technique. The ant colony optimization algorithm plays a task of feature optimization of the text data mapping for classification. Performance with all tree dataset webKB, Yahoo, Rcv1, along F1, BEP, and HLOSS, Result of classification by ACO is better as compared with RSVM AND ML-FRC algorithm.

**In [9]** they used Words sense disambiguation method and evaluate two algorithms Sequential Information Bottleneck and K means algorithm. They used 446 documents downloaded from the EMMA repository and the Document Categorized with a class label as state and purpose. The proposed methodology has shown better in cluster purity results.

**In [10],** authors present a similarity computation technique that is based on implicit links extracted from the query-log and used with K-Nearest Neighbors (KNN) in web page classification. The new computed similarity based on clicks frequencies helps enrich KNN for web page classification. This similarity uses neighborhood information and helps reduce the effect of the problem of dimensionality faced when using KNN based on the text-only. To classify a web page $pi$ , KNN calculates similarities between $pi$ and each web page in the training set. Then, it ranks web pages in the training set based on those similarities. In our case, KNN does a two-level ranking. First, it ranks web pages using the implicit links-based similarity. Then, web pages having this similarity equal to zero are ranked again using the cosine similarity.

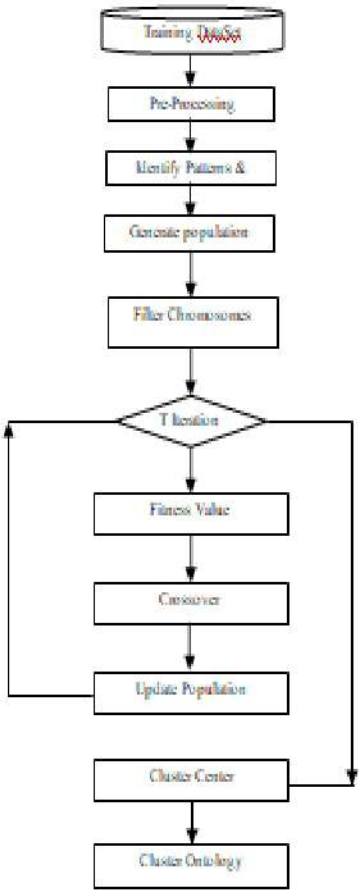
**In [11],** they aimed to develop a classifier that can categorize web pages based on their ability to attract random surfers. Web pages are classified into "bad" and "not bad" classes, where the "bad" class implies poor attention drawing ability. In the proposed approach, the web page content is divided into objects. The area occupied by these objects served as the attribute of the classifier. The experiments with various classification algorithms supported by the WEKA tool prove that two of those, namely the random subspace and the RBF networks, give high accuracy (83.33%) with high precision and recall.

## III. PROPOSED METHODOLOGY

In this work class of word document were identify by using pattern based approach where type of document representative patterns were clustered into respected class. For clustering of identified pattern genetic algorithm Fire Fly was used. For ease of understanding block diagram was shown in fig. 1. Preprocessing is initial filtration of input data where un-wanted or noisy data get remove from dataset. Here work has adopted stop-word removal technique. This approach takes each tweet in as input and remove stop words {a, an, of, the, for, to, from, in… etc.} present in it. These steps remove noise of input data so important words which give meaning to sentence are separate. This can be understand by "Ram is not happy because he loose cricket trophy", "My brother will be happier if we loose this game", so after pre-processing set of words will be: {Ram, Happy, Loose, Cricket, and Trophy}, {Brother, Happy, Loose, and Game}.

### 1. Generate Pattern

As text content classification is done by two approaches first was term and other was pattern feature. This approach use pattern feature to classify tweets where instead of assigning sentiment to a term pattern is more effective. So based on this concept patterns were identify by the successive words in a tweets. Hence successive set of words present in multiple tweets are considers as the pattern. Identification of pattern was done by:



Input T // Tweet Terms
Output: P //Pattern
1. Loop i = 1:T // For each term in Tweet
2. X←T[i]
3. Loop j = i+1:T
4. Y←T[j]
5. V←Intersect(X, Y)
6. If V size more than 2 words
7. P←V
8. EndIF
9. EndLoop
10. EndLoop

**2. Normalize Feature Vector**

As keywords are fetch from each document is collect in single vector UWV, which is term as Unique Word Vector. In UWV all set of documents send their keyword list which collect unique words from each document and counter of words are also maintain which was sum of all document presence. In this step once this list of UWV was prepare than final feature vector of each document was prepared which was collectively represent in form of matrix where each row is a document and each column represent a word from UWV, while presence of any keyword of a document is a non zero value. In order to normalize this vector in scale of 0-1 each row is divided by its corresponding summation. This help algorithm to identify the important words of a vector or important word having high value plays decision role.

**3. Generate Population**

Collection of set of cluster center were generate in form of chromosome which act as probable solution is termed as population. So in this step probable set of solution was developed randomly. Here each cluster center is document patterns which work as cluster center. So PP is an matrix of represent population while each row work has n number of cluster center for Cn clusters, as per number of sentiment. Now if PP have m column than Eq. 1 presents the population as:

PP←Random(P,Cn, m)

**4. Light Intensity of Pattern**

Calculation of this was done by estimating the total presence of pattern in available dataset. So as per pattern presence in dataset intensity value was set.

$$I_p = P_r \times e^{-\tau r}$$

Where $I_p$ is intensity of $P^{th}$ pattern, $P_r$ is presence ratio of $p^{th}$ pattern in the dataset. While is constant value range between 0-1 and r is random number vary from 0-1 for each pattern.

**5. Fitness Function**

In this step fitness value of each chromosome were evaluate by estimating the difference from the cluster center for other set of patterns (non cluster center patterns). Here paper has involved graph weigh for estimating the difference between pattern with intensity of the corresponding cluster center.

$$F_m = \sum_{i=1}^{P} Min(W_{j,})_{j=1}^{n} \times I_{j'}$$

In above equation Fm is fitness value of mth chromosome and W is weight value between two pattern so if chromosome have n cluster than assigning pattern P is send to minimum weight value cluster. j' is selected cluster center having minimum weight.

**6. Crossover**

In this work population PP chromosome values were modified by best chromosome patterns as per random position. Here best solution change other set of solutions at different cluster center position. This crossover generates other set of solution which evaluate and compared with previous fitness value.

**7. Cluster Center**

So above steps of fitness value evaluation, crossover and population uptation done iteratively for T times. Hence after T number of iteration solution gives cluster center for each type of document were identified. This selection of final cluster center depends on fitness value of population obtained.

**8. Cluster Document**

Once iteration of algorithm was over than proposed work get final cluster center document set which can be known as best chromosome in the available population. Here as per obtained cluster center each non-centroid document is cluster into respected class of document.

**9. Proposed Algorithm:**

Input: DS Document Dataset, Cn Cluster Number
Output: DC // Document Classified

1. DS ←Stop-Word-Removal(DS) // Here Stop words are remove from the Input text file
2. K←Fetch-Keywords(DS) // Here Keywords are retrieve from each text file.
3. FV←Normalize-Feature-Vector(DS)// FV: Document numeric feature vector
4. PP←Random(P, Cn, m)
5. I←Intensity(FV,P)
6. Loop 1:T
7. F←Fitness_Function(PP, FV)
8. PP←Crossover(F,PP)
9. I←Intensity_Updation(P, F)
10. EndLoop
11. FCC←Cluster_Center(PP, FV) // FCC: Final Cluster Center
12. Loop1:TD // For Each Tweet
13. DC←Cluster_Document(FCC, DS)
14. EndLoop

In above algorithm DS document dataset which was collection of text files and number of cluster center were pass as input. While output was DC document classified

© 2020 IJSRET

where each input DS text files were grouped in any of Cn cluster.

## IV. EXPERIMENTS & RESULTS ANALYSIS

Implementation of proposed genetic algorithm based document clustering approach model was done on MATLAB software because of collection of number of inbuilt function such as textscan to separate string into words, reading writing of text files, comparison of word, collection of words into structure, etc.

**Dataset**

In this work experiment is done on actual collection of text files dataset content obtained from various resources of journals where three classes of documents were collect for clustering. Table 1 shows explanation of each class of document.

Table I: Experimental dataset explanation.

| Document Type | | Count |
|---|---|---|
| Set 1 | Digital Communication | 12 |
| Set 2 | Computer Science | 12 |
| Set 3 | Electrical Load Balance | 12 |

**Evaluation Parameter**

$$\Pr ecision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$\mathrm{Re}\,call = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F - Score = \frac{2 * \mathrm{Re}\,call * \Pr ecision}{\mathrm{Re}\,call + \Pr ecision}$$

$$Accuracy = \frac{Correct\_Classification}{Correct\_Classification + Incorrect\_Classification}$$

**Result**

Existing work done in UFCGA [11] was used to compare the proposed fire fly document clustering genetic algorithm.

Table II: Digital Communication Document Class Result Comparison.

| Parameters | UCGA [11] | FFDC |
|---|---|---|
| Precision | 0.6923 | 0.8333 |
| Recall | 0.75 | 1 |
| F-Measure | 0.72 | 0.9091 |
| Accuracy | 80.56 | 83.33 |

Above table 2 shows that proposed work FFDC has improved the evaluation parameters values as compared to previous work UCGA. Use of fire fly with normalization of features has improved the work accuracy. Pattern based document clustering improve clustering efficiency of proposed FFDC algorithm for digital communication class of documents.

Table III: Computer Science Document Class Result Comparison.

| Parameters | UFCGA [11] | FFDC |
|---|---|---|
| Precision | 0.6667 | 1 |
| Recall | 0.6667 | 0.75 |
| F-Measure | 0.6667 | 0.857 |
| Accuracy | 80.56 | 83.33 |

Above table 3 shows that proposed work FFDC has improved the evaluation parameters values as compared to previous work UCGA for computer science field research documents.

Use of fire fly with normalization of features has improved the work accuracy. Pattern based document clustering improve clustering efficiency of proposed FFDC algorithm for computer science class of documents.

Table IV: Electrical Load Balance Document Class Result Comparison.

| Parameters | UFCGA [11] | FFDC |
|---|---|---|
| Precision | 0.7273 | 0.8333 |
| Recall | 0.6667 | 1 |
| F-Measure | 0.6957 | 0.9091 |
| Accuracy | 77.78 | 80 |

Above table 4 shows that proposed work FFDC has improved the evaluation parameters values as compared to previous work UCGA for electrical load balance field research documents.

Use of fire fly with normalization of features has improved the work accuracy. Pattern based document clustering improve clustering efficiency of proposed FFDC algorithm for electrical load balance class of documents.

Below Fig. 2 and 3 shows that proposed work FFDC has improved the average precision and accuracy parameters values as compared to previous work UCGA for each field research documents.

© 2020 IJSRET

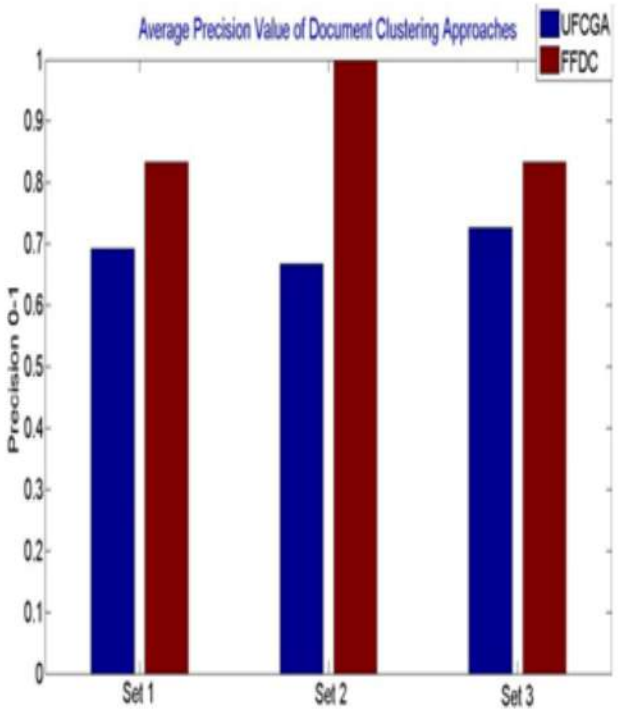



Fig.2. Average Precision value based comparison.

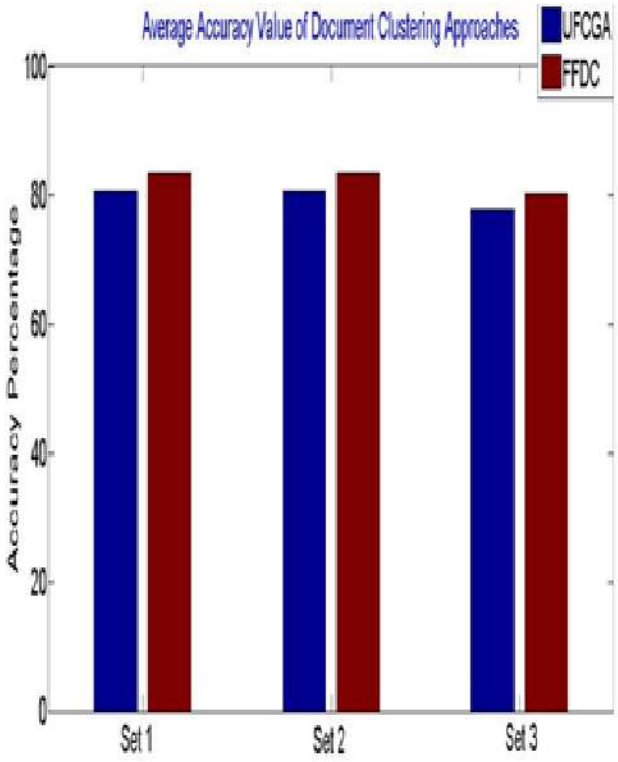



Fig.3. Average Precision value based comparison.

Use of fire fly with normalization of features has improved the work accuracy. Pattern based document clustering improve clustering efficiency of proposed FFDC algorithm for each class of documents.

## V. CONCLUSIONS

Researchers are publishing number of paper in this digital world, so gathering of relevant paper or assigning relevant paper to particular field reviewer is an important issue. This paper has resolved this issue of identifying the research paper class as per content. As text data is unorganized type of data so fetching a feature from it plays an important role in classification. Hence this work has utilized the pattern feature form the document. Use of fire fly genetic algorithm for clustering has involved un-supervised clustering where prior information or any format of document is not required. Experiment was done on real dataset having different type of data files. Results shows that proposed work has improved the precision value by 21.76% while accuracy of document classification as also improved by 3.146%. In future researcher can introduce some learning model to increase the accuracy of work as well.

## REFERENCES

[1]. Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for textdocuments classification. Journal of Advances in Information Technology, 1, 4-20.

[2]. Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for textdocuments classification. Journal of Advances in Information Technology, 1, 4-20.

[3]. Brindha, S., Sukumaran, S., & Prabha, K. (2016). A survey on classification techniques for text mining. Proceedings of the 3rd International Conference on Advanced Computing and Communication Systems. IEEE. Coimbatore, India.

[4]. K. Sarkar and R. Law, ``A novel approach to document classi_cation using WordNet," CoRR, vol. 1, pp. 259_267, Oct. 2015. [Online]. Available: https://arxiv.org/abs/1510.02755

[5]. Vasa, K. (2016). Text classification through statistical and machine learning methods: A survey. International Journal of Engineering Development and Research, 4, 655-658.

[6]. Abroyan N. Convolutional and recurrent neural networks for real-time data classification. innovative Computing Technology (INTECH), 2017 Seventh International Conference on 2017 Aug 16 (pp. 42-45). IEEE.

[7]. Zhang Y, Er MJ, Venkatesan R, Wang N, Pratama M. Sentiment classification using comprehensive attention recurrent models. neural Networks (IJCNN), 2016 International Joint Conference on 2016 Jul 24 (pp. 1562-1569). IEEE.

[8]. Nema, Puneet, and Vivek Sharma. "Multi-label text categorization based on feature optimization

using ant colony optimization and relevance clustering technique." Computers, Communications, and Systems (ICCCS), International Conference on. IEEE, 2015.

[9]. Kulathunga, Chalitha, and D. D. Karunaratne. "An ontology-based and domain-specific clustering methodology for financial documents", advances in ICT for Emerging Regions (ICTER), 2017 Seventeenth International Conference on. IEEE, 2017.

A. Belmouhcine et M. Benkhalifa, « Implicit Links-Based Techniques to Enrich K-Nearest Neighbors and Naive Bayes Algorithms for Web Page Classification », in Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015, vol. 403, R. Burduk, K. Jackowski, M. Kurzyński, M. Woźniak, et A. Żołnierek, Éd. Cham: Springer International Publishing, 2016, p. 755 766.

[10]. G. Khade, S. Kumar, et S. Bhattacharya, « Classification of web pages on attractiveness: A supervised learning approach », in Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on, 2012, p. 1–5.

[11]. Alan Díaz-Manríquez , Ana Bertha Ríos-Alvarado, José Hugo Barrón-Zambrano, Tania Yukary Guerrero-Melendez, And Juan Carlos Elizondo-Leal. "An Automatic Document Classifier System Based on Genetic Algorithm and Taxonomy". accepted March 9, 2018, date of publication March 15, 2018, date of current version May 9, 2018.

© 2020 IJSRET