

# Performance Analysis of Modified Shuffled Frog Leaping Algorithm for Multi-document Summarization Problem

Rasmita Rautray, Rasmita Dash and Rajashree Dash

Dept. of Computer Science & Engineering, Siksha 'O' Anusandhan Deemed to be University, Odisha, India

E-mail: rashmitarouray@soa.ac.in, rasmitadash@soa.ac.in, rajashreedash@soa.ac.in

**Keywords:** multi-document summarization, extractive summary, modified shuffled frog leaping algorithm

**Received:** April 23, 2018

*Due to massive growth of Web information, handling useful information has become a challenging issue in now-a-days. In the past few decades, text summarization is considered as one of the solution to obtain relevant information from extensive collection of information. In this paper, a novel approach using modified shuffled frog leaping algorithm (MSFLA) to extract the important sentence from multiple documents is presented. The effectiveness of MSFLA algorithm for summarization model is evaluated by comparing the ROUGE score and statistical analysis of the model with respect to results of other summarization models. The models are demonstrated by the simulation results over DUC datasets. In the present work, it elucidates that MSFLA based model improves the results and find advisable solution for summary extraction.*

*Povzetek: Na bazah dokumentov je bil uporabljen nov algoritem MSFLA za generiranje povzetkov dokumentov.*

## 1 Introduction

Present days, growing of information exponentially in Web initiates information overload problem. As a result automatic text summarization (TS) has increasing value to various real-world applications. TS summarize information from single or multiple documents which share an explicit or implicit main topic. It facilitates users to quickly catch the most relevant and important information through large text data collections. Shortening of a large text document or documents into a concise form is called as single or multi document summarization respectively. As multi-document summarization processes multiple documents, therefore search space of multi-document summarization is large and considered as an enlargement of single document summarization. Hence it makes more challenging for selection of important sentences. In that context, summarization of multiple documents can be recognized as an optimization problem with the objective of producing optimal summary containing relevant and informative sentences of the original input documents [1].

As a challenging issue for text mining, automatic document summarization had been well-studied during half a century years [2]. The great majority of numerous approaches developed are extraction-based, which produce a summary using only existing sentences (or text fragments) extracted from the original text, so they are conceptually simple and more practicable compared to abstractive methods, which attempt to reproduce sentence by using complicated natural language generation techniques such as sentence compression [3] information fusion [4] and reformulation [5].

In literature, many evolutionary computation based contributions have been suggested for sentence selection

from huge collection of information. The domain is already rich with the proposal of many evolutionary based summarization models development such as (GA), differentia evolution (DE), particle swarm optimization (PSO), harmony search (HS), cat swarm optimization (CSO), cuckoo search (CS) etc. In few cases these techniques suffers from premature convergence and gets trapped into its local optima. The focuses have to be exploitation and exploration abilities of these evolutionary algorithms. Further to improve the performance motivations have to be made on population diversity in the progressive procedure and a sophisticated approach for information sharing among each participant in the distribution. To overcome these issues an evolutionary approach called Shuffled frog leaping algorithm (SFLA) is proposed.

There are many applications in which shuffle frog leaping algorithm is applied. A detail description is highlighted as follows.

Tarun et.al. [6] applied opposition based leaning to improve the global search of SFL algorithm. It not only improves the local search but also improves its diversity. This technique is experimented on 10 benchmark optimization function, 10 shifted function (from CEC2005) and on cost management problem in cellular network.

Dalavi et.al [7] proposed a modified SFL algorithm for hole making operation in plate of ejector mould. Many kinds of machining operations are required for hole making operation. Implementing this technique the optimal sequence of operation is identified minimizing cost for hole making operation.

An improved SFL algorithm (ISFL) is proposed by Dash et.al. [8] for currency rate prediction problem. In this technique the author has tried to improve the local and global search process. This technique introduces a new frog leaping rule with the acceleration factor and leaping inertia components. The experimental result is compared with simple SFL algorithm and based PSO and the result shows the superiority of ISFL algorithm in terms of both convergence rate and predictor accuracy.

In [9], Kaur et.al. applied an augmented shuffle frog leaping (ASFLA) approach for resources provisioning work flow scheduling infrastructure as service cloud computing environment. For task scheduling ASFLA tries to minimize the execution time and transfer time among dependent task. This approach is compared with simple SFL and PSO and a significant improvement is observed using ASFLA.

A simulated shuffle frog leaping algorithm (SSFLA) is suggested by Amirian et.al [10] for grey scale project selection scheduling in tri-objective grey environment. Implementing time limit, budget constraint and multiple objectives, a modified grey shuffle frog leaping algorithm is proposed. This technique is compared with non-dominated sorting genetic algorithm (NSGA-II) and multi objective PSO to solve this NP hard problem.

In [11], the author has stated that SFLA is a novel meta-heuristic approach, applied in many combination problem. However in continuous optimization problem the algorithm falls in local optima. Thus author has proposed a variant of SFL called levy flight based shuffle frog leaping algorithm. The effectiveness of this technique is explored using 30 benchmark function and six continuous optimization functions. As a stochastic search based learning technique, Sharma et.al.[12], has suggested a variant of shuffle frog leaping algorithm called as centroid mutated SFLA for both discrete and continuous optimization problem.

Bhattacharjee et.al.[13] proposed a modified discrete shuffle frog leaping algorithm for 01 knapsack problem. This technique is extensively investigated taking considering large number of experimental studies. Due to its discretization property, performance shows a remarkable growth for small as well as medium sized knapsack problem and as an alternative solution for large knapsack problem.

Inspired from the successful implementation of SFLA in many application areas as an optimization approach, in this study a novel Modified Shuffled frog leaping algorithm based multi document summarizer is presented. MSFLA aims to maximize content coverage criteria while reducing redundancy and preserving length of the summary. The effectiveness of the proposed model has been evaluated with respect to particle swarm optimization (PSO), cuckoo search (CS) and standard shuffled frog leaping algorithm over DUC datasets. From the experimental results, it is clearly observed that MSFLA based multi document summarizer outperforms than conventional PSO, CS and SFLA based summarizer.

The rest of the paper is organized as follows. The detail of Shuffled frog leaping algorithm is presented in Section 2. In Section 3, Modified Shuffled frog leaping

algorithm has been discussed. A detail framework for multi-document extractive summarization using MSFLA is presented in Section 4. The summary evaluation criteria and empirical study for performance analysis are discussed in Section 5 and 6 respectively. Finally conclusion is highlighted in Section 6.

## 2 Shuffled frog leaping algorithm

The Shuffled Frog Leaping Algorithm (SFLA) is a recent population based meta-heuristic algorithm. It incorporates both the benefits of memetic algorithm and social behavior of particle swarm optimization (PSO) algorithm. In SFLA (Figure 1), the population is a group of frogs, which are seeking for best available foods using search guidelines related to PSO algorithm. With the aim to search for food, the entire search process can be carried out by alternating communication of frogs in intra-cluster and inter-cluster. The intra-cluster communication performs within a memplex for local invention and the inter-cluster communication performs between the frogs belongs to different memplexes for global exploration. In traditional SFLA, assume that the initial population is generated randomly of P solutions (frogs). For each individual frog, evaluate fitness value. Afterward, sort the P in descending order of their fitness value. Then entire frogs are distributed into M number of memplexes and each memplex contains N frogs. In this process, the distribution of frogs is done in such a way that the first frog goes to first memplex, second goes to first memplex, frog M goes to M<sup>th</sup> memplex, and frog M+1 goes to first memplex, and so on till the last frog [14,15].

Within each memplex, the best and the worst frog according to their fitness are represented as  $X_b$  and  $X_w$ . The frog having global fitness is denoted as  $X_g$ . The location of worst frog is updated either based on location of local best frog or global best frog or randomly to a position, so that the frogs can move towards the optimal solution. The updating measure is as follows:

$$X_w(new) = X_w + Stp \quad (1)$$

$$Stp = rand() \times (X_b - X_w) - Stp_{max} \leq Stp \leq Stp_{max} \quad (2)$$

Where  $stp$  is frog leaping step size of range [-1, 1],  $rand()$  is a random number between [0, 1]. If  $X_w(new)$  produced a better solution, it replaces  $X_w$ . Otherwise, the calculation in Equation (1) and (2) are repeated by replacing  $X_b$  with  $X_g$ . If there is no improvement in such situation, then a randomly generated new solution is replaces to  $X_w$ . Thereafter all the memplexes are shuffled together to exchange information and generate new population for next search space.

## 3 Modified SFLA

Even though the advantage of traditional SFLA such as its simple structure, fewer number of controlling parameters and simple realization of algorithm, the algorithm have some limitations. In traditional SFLA, initial generated population is not uniform. Due to that reason diversity and searching ability of population decreases, and in local

searching process only it updates worst solution without updating the best solution. Therefore it has a negative influence on convergence speed of algorithm and solving precision. Hence to improve the performance of SFLA, a modified version of SFLA (MSFLA) is suggested in this work [8,16-18]. In MSFLA, the frog leaping step size is controlled by inserting search learning coefficient  $S$  with inertia component to change present movement status of frog during local search. The leaping rule discussed in equation 1 and 2 is modified and presented in equation 3 and 4.

$$X_w(new) = X_w + Stp \tag{3}$$

$$Stp = \phi \times Stp^{t-1} + S \times rand() \times (X_b - X_w) - Stp_{max,max} \tag{4}$$

Where,  $S$  is search learning coefficient (i.e.  $S$  is any constant but greater than one) to scale the frog's step size during local search process. If  $S$  contains a big value, leads premature convergence or results in missing of the local search producing only random search with little improvement. In equation 3 and 4, the leaping step size  $Stp$  at any instance  $t$  depends on frog's leaping step size with worst position in previous iteration. The objective of introducing inertia weight  $\phi$  is to balance in the search process. It assumes that the greater inertia weight offers exploration while a smaller one raises the local

exploration. Instead of considering a fixed inertia weight value, it is decreased repeatedly from a greater to smaller specified value [19, 20].

## 4 MSFLA framework for multi-document extractive summarization

### 4.1 Modeling the summarization problem as MSFLA problem

In order to model the summarization problem as MSFLA problem, each sentence is represented as a frog. Originally the source input  $SI$  contain number of individual documents i.e.,  $SI = \{DC_1, DC_2, K, DC_m\}$  and each individual document  $DC$  is segmented separately as  $DC = \{S_1, S_2, K, S_n\}$ , where  $m$  and  $n$  represents the number of documents and sentences in each document respectively. The number of currently available resources or sentences in the input document to the optimization problem determines the search space that allowed the frog to move in search of food or important sentence selection. In this problem, the weight of each sentence is represented

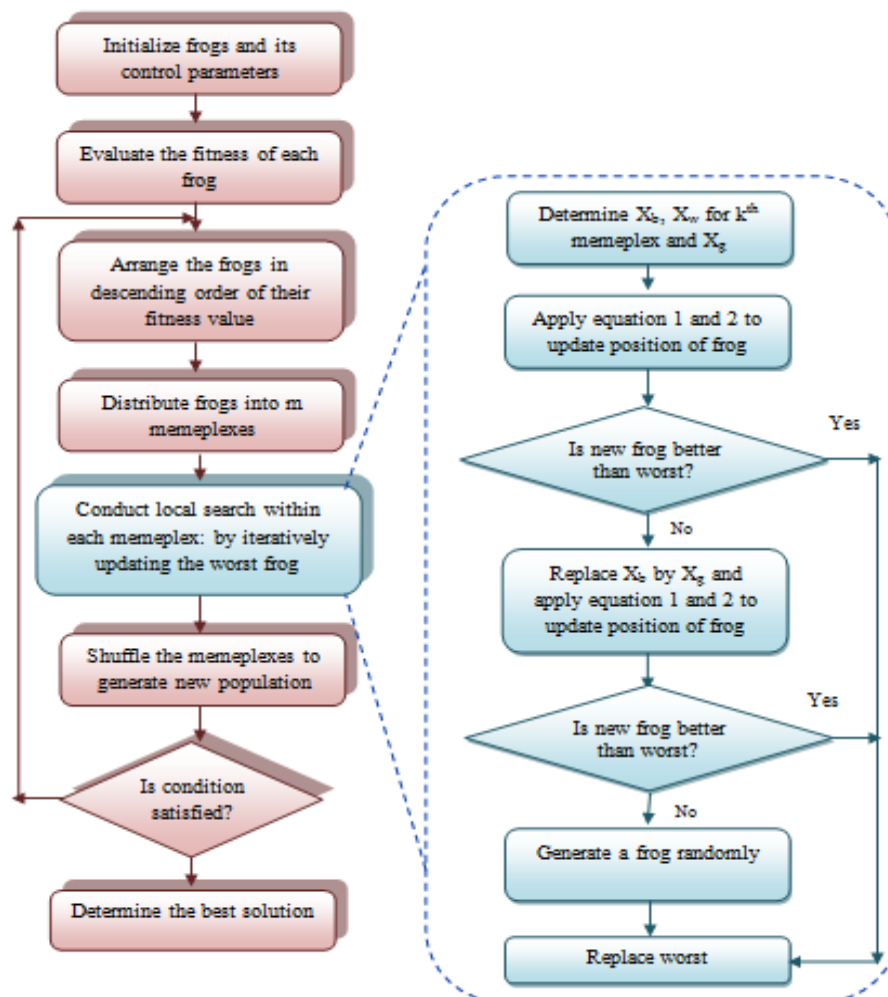


Figure 1: Flow chart of SFLA.

as the position of frog and it needs to be optimized to

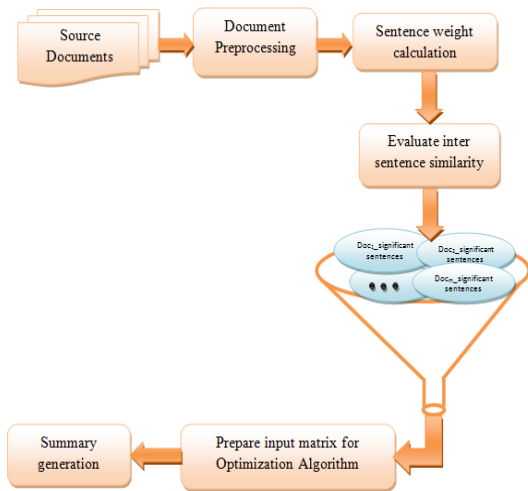


Figure 2: Proposed document summarization framework.

improve the searching process.

### 4.2 Overview of proposed summarization approach

The proposed framework of document summarization is illustrated in Figure 2. Primary objective of this approach is to generate a document in summarized form, from a set of input documents. First of all, for each document, sentences are segmented and each term of that sentence is tokenized followed by removal of stop word and stemming. Next, modified term frequency and inverse document frequency ( $MTfIdf$ ) is applied to compute weight of each sentence ( $Wt_{sen}$ ). For evaluation of inter sentence similarity, the most widely used cosine similarity metric is used. Once the similarity matrix is created, a similarity threshold is employed to select significant sentences from each document. Thereafter all significant sentences are merged into a document and their respective sentence weights are used to prepare input matrix for optimization algorithm. Finally, by comparing with summary threshold the top weighted sentences are selected to generate summary.

### 4.3 Detailed steps of proposed approach using MSFLA

- Step 1: Set input documents  $SI$ , where  $SI = \{DC_1, DC_2, K, DC_m\}$ . Each  $DC_i$  represents individual document of set  $SI$ . Each  $DC_i$  is represented in terms of number of sentences.
- Step 2: Preprocess each text document  $DC_i$  through the following sub-process.
  - Sentence segmentation: Read the text document and represent as segmented sentence  $S$  individually.

Tokenization: Read the sentence and terms of each sentence are tokenized as  $TM = \{tm_1, tm_2, K, tm_p\}$ , where  $tm_k$  for  $k=1, 2, \dots, p$ .

Stop word removal: The word which has less important significance with respect to the document is removed such as 'a', 'an', 'the' etc.

Stemming: Eliminate suffix part of the word into its common base form.

- Step 3: Calculate the sentence weight ( $Wt_{sen}$ ) for each sentence  $S_j$  of the preprocessed document  $DC_i$  using weighted sum of modified weighting scheme of term frequency and inverse document frequency [21], illustrated in equation (5).

$$Wt_{sen} = MTfIdf_{tm,DC} = MTf_{tm,DC} \times Idf_{tm} \tag{5}$$

$$MTf_{tm,DC} = \frac{Tf_{tm,DC} \times \log \frac{\sqrt{Tc}}{Tt}}{\log \left[ \left( \sum_{tm=1}^p Tf_{tm,DC}^2 \right) \times \left( \frac{DC_{len}^2}{\sqrt{Tc}} \right) \right]} \tag{6}$$

Where  $Tt = \sum_{DC=1}^{SI} Tf_{tm,DC}$       $Tf_{tm,DC} > 0$

and  $Tc = \sum_{DC=1}^{SI} \sum_{tm} Tf_{tm,DC}$

Where  $Tf_{tm,DC}$  – occurrence of term  $tm$  in document  $DC$   
 $Tt$  – raw frequency of term  $tm$  with respect to all document collection  
 $Tc$  – number of distinct terms in the document collection  
 $DC_{len}$  – length of the document  $DC$

$$Idf_{tm} = \log \left( \frac{SI}{DF_{tm}} \right) + 1 \tag{7}$$

$$DF_{tm} = \sum_{DC=1}^{SI} \begin{cases} 1 & tm \in DC \\ 0 & tm \notin DC \end{cases}$$

where  $SI$  = Total number of documents

- Step 4: Evaluate sentence-sentence similarity for the preprocessed document  $DC_i$  using cosine similarity metric.
- Step 5: For each  $DC_i$ , select least similar sentences based on a similarity threshold value.
- Step 6: Select all least similar sentences of each  $DC_i$  and merge to represent as a single document  $DC_{input}$ .
- Step 7: Set sentence weight of  $DC_{input}$  as frog information to the MSFLA.
- Step 8: Evaluate fitness value of each frog.
- Step 9: Arrange the frogs in decreasing order of their fitness value.
- Step 10: Distribute entire frogs into  $M$  number of memplexes and each memplex contains  $N$  frogs such that  $P = M \times N$ . In this process, the distribution of frogs is done in such a way that the first frog goes to first memplex, second goes to first memplex, frog  $M$  goes to  $M^{th}$  memplex, and frog  $M+1$  goes to first memplex, and so on.
- Step 11: For each memplex, do step 12.
- Step 12: Local search (until iterative steps for each memplex is not reached):
  - Step 12.1: Within each memplex, determine the frog with best and worst position such as  $X_b$  and  $X_w$ . determine the frog with global best position  $X_g$  with respect to entire frog population.

*Step 12.2: Evaluate new position of the worst frog by exchanging information within memplex by using equation (1).*

*Step 12.3: If fitness value of  $X_w$ (new) is better than current one,  $X_w$  is replaced by the new one. Go to step 12.*

*Step 12.4: Otherwise, evaluate new position of the worst frog by exchanging information between memplex using equation (1). But in this case,  $X_b$  is replaced by  $X_g$  in equation (1).*

*Step 12.5: If fitness value of  $X_w$ (new) is better than current one,  $X_w$  is replaced by the new one. Go to step 12.*

*Step 12.6: Otherwise, if resulting leap does not produced any improvement of the worst frog, then new position is generated randomly. Go to step 12.*

*Step 13: Shuffle the frogs of all memplexes as new frog population.*

*Step 14: The parameters  $S$  and  $\phi$  are adapted at  $t$  number iteration as follows:*

$$S(t) = S_{low} + (S_{high} - S_{low}) \times t / \text{total number of iteration} \quad (8)$$

$$\phi(t) = \phi_{low} + (\phi_{high} - \phi_{low}) \times t / \text{total number of iteration} \quad (9)$$

*Step 15: Finally, the frog with best fitness value is considered as candidate summary sentences.*

*Step 16: Select summary sentences chronologically from the document set based on their threshold by comparing with candidate summary sentences.*

## 5 Summary evaluation criteria

The objective function  $f$  of proposed model is prepared in such a way that it maximizes coverage criteria while reducing redundancy and preserving length of the summary. Therefore, the authors have tried to form a summary from a set of documents with the objective of content coverage, non-redundancy and length. The summarization problem can be formalized as follows:

$$f = \sum_{i=1}^{m-1} \sum_{j=i+1}^m \left[ \begin{array}{l} Sim(DC, S_i) + Sim(DC, S_j) \\ - Sim(S_i, S_j) \end{array} \right] x_{ij}, \quad (10)$$

$$\text{such that } len(S_i) \text{ and } len(S_j) \leq L, \quad (11)$$

$$x_{ij} \in [0,1] \quad \forall i, j$$

$$Sim(S_i, S_j) = \frac{\sum_{k=1}^p w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^p w_{ik}^2 \cdot \sum_{k=1}^p w_{jk}^2}}, \quad (12)$$

$$i, j = 1, 2, K, n$$

where  $w_{ik}$  or  $w_{jk}$  is weightingscheme assigned to a term

In equation (10), the first two term guarantees that relevance of the summary and it covers main content of the document set. Whereas, the third term avoid multiple textual units, that convey the same information. The length of the summary is bounded by cardinality constraint, is discussed in equation (11).

## 6 Empirical study

This section introduces: 1) detail of benchmark dataset for evaluation of summary; 2) setting of controlling parameters during simulation; 3) describe the metrics to assess performance of the summary; 4) performance analysis.

### 6.1 Evaluation setup on the benchmark dataset

For the evaluation of proposed framework, the datasets from Document Understanding Conference (DUC) is used [22]. It is a benchmark data corpus for text summarization problem. This contains input documents along with few reference summaries. The DUC datasets i.e., DUC2006 and DUC2007 are distributed through ACQUINT, and used for this experimental study. DUC2006 and DUC2007 contains 50 and 45 different document clusters, each cluster contains 25 independent documents respectively. Each document cluster contains reference summaries of 250 words, which answers the question(s) in the topic statement.

### 6.2 Parameter setup

The setup of control parameters of any optimization algorithm is application oriented and no fixed value is assign to these parameters. Therefore derivation of parameters is obtained through number of simulations. In this study, the experimentation has been conducted taking the population size 50 to 200 and fixing the number of memplex 10. Initially the memplex are upgraded with 10 number of iteration. However, after 8 iterations no such remarkable upgradation has been observed. Here the author has added the optimal environment setup for considered application. The controlling parameters of MSFLA algorithm such as: size of population (SP) is 160, memplex size (m) is 10, iterative steps for memplex (im) are 8, number of iteration (it) is 50, search learning coefficient and inertia weight are decreasing value of range  $[1, 1 + \text{rand}()]$  and  $[0, 1]$  respectively.

### 6.3 Performance evaluation metrics

The comprehensive performance study of MSFLA based summarizer has been conducted over DUC dataset and evaluated by ROUGE [23, 24]. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is used as the official evaluation metric for text summarization. A set of metrics such as ROUGE-L, ROUGE-N, ROUGE-S, ROUGE-W and ROUGE-SU are available in ROUGE to evaluate system generated summary against a set of reference summaries. In this study, ROUGE-N metric is used to match between system generated summaries and reference summaries.

$$ROUGE - N = \frac{\sum_{Sen \in Reference\_summary} \sum_{N-gram \in Sen} Count_{match}(N-gram)}{\sum_{Sen \in Reference\_summary} \sum_{N-gram \in Sen} Count(N-gram)} \quad (13)$$

Where N is contiguous sequence of terms, count match is the highest number of N-grams co-occurring in

system summary and reference-summaries. Count is the number of contiguous sequence of N-terms in the reference summaries.

### 6.4 Result analysis

The summary result has been evaluated by ROUGE-N score with N is 1 and 2 i.e. ROUGE-1 (R1) and ROUGE-2 (R2) metrics. ROUGE-1 and ROUGE-2 refers to word wise comparison and two words comparison at a time between the system summary and the reference summaries. Based on content coverage, anti-redundancy and length of summary ROUGE-N is calculated. A model producing higher ROUGE score reveals higher similarity

of the system generated summary with respect to the original input document sets. Table 2 shows the ROUGE score obtained by proposed summarization model. ROUGE-1 score of all summarizer are falling within the range 0.41 to 0.44 and with respect to ROUGE-2 it is within the range 0.07 to 0.16 for DUC 2006 and DUC2007 dataset respectively. The statistical analysis in term of min (Min), average (Avg) and standard deviation (Std) are observed in Table 2 and Table 3 out of 20 independent runs for the MSFLA, SFLA, CS and PSO algorithm on DUC 2006 and DUC 2007 dataset respectively. In comparison with models illustrated in Table 1, it can be

Methods	DUC 2006		DUC 2007	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
PSO	0.4112	0.0784	0.4096	0.0762
CS	0.4311	0.1398	0.4243	0.1034
SFL	0.4320	0.1503	0.4317	0.1223
MSFLA	0.4408	0.1610	0.4358	0.1412

Table 1: ROUGE score of proposed summarizers on DUC2006 and DUC2007 datasets.

ROUGE metric		PSO	CS	SFL	MSFLA
ROUGE 1	Min	0.3908	0.4042	0.4016	0.4119
	Avg	0.4029	0.4142	0.4133	0.4228
	Std	0.0076	0.0087	0.0091	0.0096
ROUGE 2	Min	0.0487	0.0701	0.0811	0.1094
	Avg	0.0602	0.0890	0.1163	0.1305
	Std	0.0106	0.0210	0.0282	0.0186

Table 2: Performance comparison of proposed summarizers on DUC2006.

ROUGE metric		PSO	CS	SFL	MSFLA
ROUGE 1	Min	0.3916	0.4000	0.4053	0.4162
	Avg	0.3996	0.4119	0.4219	0.4270
	Std	0.0060	0.0074	0.0097	0.0061
ROUGE 2	Min	0.0643	0.0803	0.1011	0.1211
	Avg	0.0704	0.0944	0.1134	0.1337
	Std	0.0046	0.0082	0.0077	0.0073

Table 3: Performance comparison of proposed summarizers on DUC2007.

Methods	DUC 2006		DUC 2007	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
PSO	7.19	105.35	6.39	85.30
CS	2.25	15.16	2.71	36.55
SFL	2.03	7.11	0.94	15.45

Table 4: Relative Improvement comparison of MSFLA based summarizer and other summarizer with respect to ROUGE-1 and ROUGE-2.

realized that MSFLA based summarizer reveals the result based multi-document summarizer outperforms than other

Methods	DUC 2006		DUC 2007	
	Avg	95% CI	Avg	95% CI
PSO	0.4029	[0.399569, 0.406231]	0.3996	[0.396748, 0.402452]
CS	0.4142	[0.410387, 0.418013]	0.4119	[0.408382, 0.415418]
SFL	0.4133	[0.409312, 0.417288]	0.4219	[0.417289, 0.426511]
MSFLA	0.4228	[0.418593, 0.427007]	0.4270	[0.4241, 0.4299]

Table 5: Average (Avg) values of ROUGE-1 scores and confidential interval of methods.

Methods	DUC 2006		DUC 2007	
	Avg	95% CI	Avg	95% CI
PSO	0.0602	[0.0555, 0.0648]	0.0704	[0.068213, 0.072587]
CS	0.0890	[0.0798, 0.0982]	0.0944	[0.090502, 0.098298]
SFL	0.1163	[0.103941, 0.128659]	0.1134	[0.10974, 0.11706]
MSFLA	0.1305	[0.122348, 0.138652]	0.1337	[0.13023, 0.13717]

Table 6: Average (Avg) values of ROUGE-2 scores and confidential interval of methods.

of other summarizers on both the datasets.

Furthermore, the validity of MSFLA based summarizer is verified by obtaining relative improvement of MSFLA based summarizer over other methods in terms of ROUGE score. The relative improvement comparison is calculated using the following formula.

$$\frac{(\text{MSFLA based result} - \text{other summarizer result})}{\text{other summarizer result}} \times 100 \quad (14)$$

Where *other summarizer result* includes PSO, CS or SFLA based summarizer.

From the analysis of results it is clearly observed that, the performance of proposed method based on MSFLA based summarizer is providing consistent result for the given scenario.

Thereafter a statistical significance test has been conducted at the 5% significance level of the summarization results [25]. The average values and 95% confidence interval (CI) of ROUGE-1 and ROUGE-2 scores of each method for DUC 2006 and DUC 2007 data sets are shown in Tables 5 and 6 respectively.

From Table 5 and 6, it is observed that the average values of ROUGE-1 and ROUGE 2 for MSFLA method on all data sets are better than that for the other methods.

## 7 Conclusion

This paper attempts to present a MSFLA based multi-document summarizer. It highlights the implementation steps of MSFLA algorithm to optimize the sentence weight to generate summary from multiple documents. It is modified variation of standard SFLA. The application of PSO, CS and SFLA to extract sentences from multiple documents is also explored. Proposed summarizers are validated over DUC dataset. From the experimental analysis of ROUGE metrics and confidential interval of statistical significance test, clearly refer that the MSFLA

summarizer models discussed in this experimental study.

The future study of this research work will be extended for the abstractive summarization problem. The performance of summarizer will also be compared with other competent nature inspired algorithms.

## 8 References

- [1] Rautray, R., & Balabantaray, R. C. (2017). An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA. Applied Computing and Informatics. <http://dx.doi.org/10.1016/j.aci.2017.05.003>.
- [2] H. P. Luhn (1958). The automatic creation of literature abstracts, IBM Journal of Research and Development 2 (2), 159–165, doi:10.1147/rd.22.0159.
- [3] L. Wang, H. Raghavan, V. Castelli, R. Florian, C. Cardie, A sentence compression based framework to query-focused multi-document summarization, arXiv preprint arXiv:1606.07548.
- [4] R. Barzilay, K. R. McKeown, M. Elhadad (1999) Information fusion in the context of multi-document summarization, in: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, Association for Computational Linguistics, pp.550–557, DOI: 10.3115/1034678.1034760.
- [5] K. R. McKeown, J. L. Klavans, V. Hatzivassiloglou, R. Barzilay, E. Eskin (1999). Towards multidocument summarization by reformulation: Progress and prospects, In Proceedings of AACL-99, doi>10.1093/comjnl/35.3.268.
- [6] Sharma, T. K., & Pant, M. (2017). Opposition based learning ingrained shuffled frog-leaping algorithm.

- Journal of Computational Science, 21, 307-315, <https://doi.org/10.1016/j.jocs.2017.02.008>.
- [7] Dalavi, A. M., Pawar, P. J., & Singh, T. P. (2016). Optimal sequence of hole-making operations using particle swarm optimization and modified shuffled frog leaping algorithm. *Engineering Review*, 36(2), 187-196.
- [8] Dash, R. (2017). An improved shuffled frog leaping algorithm based evolutionary framework for currency exchange rate prediction. *Physica A: Statistical Mechanics and its Applications*, 486, 782-796, DOI: 10.1016/j.physa.2017.05.044.
- [9] Kaur, P., & Mehta, S. (2017). Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. *Journal of Parallel and Distributed Computing*, 101, 41-50, DOI: 10.1016/j.jpdc.2016.11.003.
- [10] Amirian, H., & Sahraeian, R. (2017). Solving a grey project selection scheduling using a simulated shuffled frog leaping algorithm. *Computers & Industrial Engineering*, 107, 141-149, <https://doi.org/10.1016/j.cie.2017.03.018>.
- [11] Tang, D., Yang, J., Dong, S., & Liu, Z. (2016). A lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems. *Applied Soft Computing*, 49, 641-662.
- [12] Sharma, S., Sharma, T. K., Pant, M., Rajpurohit, J., & Naruka, B. (2015). Centroid mutation embedded shuffled frog-leaping algorithm. *Procedia Computer Science*, 46, 127-134, <https://doi.org/10.1016/j.asoc.2016.09.002>.
- [13] Bhattacharjee, K. K., & Sarmah, S. P. (2014). Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. *Applied Soft Computing*, 19, 252-263, <https://doi.org/10.1016/j.asoc.2014.02.010>.
- [14] Hasanien, H. M. (2015). Shuffled frog leaping algorithm for photovoltaic model identification. *IEEE Transactions on Sustainable Energy*, 6(2), 509-515, DOI: 10.1109/TSTE.2015.2389858.
- [15] Kaur, P., & Mehta, S. (2017). Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. *Journal of Parallel and Distributed Computing*, 101, 41-50, doi>10.1016/j.jpdc.2016.11.003.
- [16] Huynh, T. H. (2008, April). A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers. In *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on* (pp. 1-6). IEEE, DOI: 10.1109/ICIT.2008.4608439.
- [17] Zhang, X., Hu, X., Cui, G., Wang, Y., & Niu, Y. (2008, June). An improved shuffled frog leaping algorithm with cognitive behavior. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on* (pp. 6197-6202). IEEE, DOI: 10.1109/WCICA.2008.4592798.
- [18] Pu, H., Zhen, Z., & Wang, D. (2011). Modified shuffled frog leaping algorithm for optimization of UAV flight controller. *International Journal of Intelligent Computing and Cybernetics*, 4(1), 25-39, <https://doi.org/10.1108/17563781111115778>.
- [19] Chittineni, S., Godavarthi, D., Pradeep, A. N. S., Satapathy, S. C., & Reddy, P. P. (2011, July). A modified and efficient shuffled frog leaping algorithm (MSFLA) for unsupervised data clustering. In *International Conference on Advances in Computing and Communications* (pp. 543-551). Springer, Berlin, Heidelberg.
- [20] Liang, B., Zhen, Z., & Jiang, J. (2016). Modified shuffled frog leaping algorithm optimized control for air-breathing hypersonic flight vehicle. *International Journal of Advanced Robotic Systems*, 13(6), <https://doi.org/10.1177/1729881416678136>.
- [21] Sabbah, T., Selamat, A., Selamat, M. H., Al-Anzi, F. S., Viedma, E. H., Krejcar, O., & Fujita, H. (2017). Modified frequency-based term weighting schemes for text classification. *Applied Soft Computing*, 58, 193-206, DOI: 10.1016/j.asoc.2017.04.069.
- [22] <http://duc.nist.gov>
- [23] C. Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 71-78). Association for Computational Linguistics, DOI: 10.3115/1073445.1073465.
- [24] Rautray, R., & Balabantaray, R. C. (2017). Cat swarm optimization based evolutionary framework for multi document summarization. *Physica A: Statistical Mechanics and its Applications*, 477, 174-186, DOI: 10.1016/j.physa.2017.02.056.
- [25] Hollander, M., & Wolfe, D. A. (1999). *Nonparametric statistical methods* (2nd ed.). Wiley-Interscience (p.787), DOI:10.1002/9781119196037.