Research Article

Brainstorm optimization for multi-document summarization

Tamilselvan Jayaraman [1]* Dr.A.Senthilrajan[2]

[1,2]Department of Computational Logistics, Alagappa University, Karaikudi, Taminadu, India

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Document summarization is one of the solutions to mine the appropriate information from a huge number of documents. In this study, brainstorm optimization (BSO) based multi-document summarizer (MDSBSO) is proposed to solve the problem of multi-document summarization. The proposed MDSBSO is compared with two other multi-document summarization algorithms including particle swarm optimization (PSO) and bacterial foraging optimization (BFO). To evaluate the performance of proposed multi-document summarizer, two well-known benchmark document understanding conference (DUC) datasets are used. Performances of the compared algorithms are evaluated using ROUGE evaluation metrics. The experimental analysis clearly exposes that the proposed MDSBSO summarization algorithm produces significant enhancement when compared with the other summarization algorithms.

Keywords: Multi-Document Summarization, Particle Swarm Optimization, Bacterial Foraging Optimization, Brain Storm Optimization.

1. Introduction

Document summarization is the process of making a shorter version of the original text without dropping any content from the given document. The summary will help the reader to make a decision about the documents whether it is significant or not [1]. The task of summarization is done in two ways such as extractive and abstractive. An extractive summary will extract significant parts such as paragraphs, sentences, etc. An abstractive summary uses linguistic investigation to make a summary [2].

The document summarization is classified into two types based on the number of documents such as single document summarization and multi-document summarization. The single document summarization compresses a given single document to a shorter version. The multi-document summarization process aimed at extraction of information from multiple document sources. The multi-document summarization is a challenging task when compared with a single document summarization due to large search space in multi-documents. The problem of multi-document summarization is accepted as optimization problem. The main aim of the multi-document summarization problem is to generate best possible informative summary of the original documents.

The multi-document summarization problem is solved by using many techniques including classification [3], clustering [4] and regression [5]. The various nature inspired optimization techniques are applied to solve both single and multi-documents summarization including genetic algorithm (GA) [6], differential evolution (DE) [7], particle swarm optimization (PSO) [8, 9] ant colony optimization (ACO) [10], cuckoo search optimization (CSO) [11], firefly algorithm (FA) [12], krill herd (KH) [13] and bacterial foraging optimization (BFO) [14], social spider optimization (SSO) [15], cat swarm optimization (CSO) [16]. However, these kinds of nature inspired optimization algorithms results in poor balance between exploration and exploitation [17]. On the other hand, BSO is a talented swarm intelligence (SI) algorithm proposed by Yuhui Shi [18]. The BSO algorithm is more attractive to the researcher because of its efficiency and simplicity. The key ideas of the BSO are mutation and clustering which is encouraging in searching ability to find global optimum and preserving population diversity. The BSO algorithm is applied to solve many real-world applications including data classification [19, 20], multi-objective optimization problem [21], hierarchical clustering analysis [22], multi-strategy BSO for global optimization functions [23], image classification [24], hardware / software partitioning [25], renewal energy system [26].

In this research work, BSO algorithm is proposed for solving multi-document summarization problem (MDSBSO). The performance of proposed multi-document summarization algorithm is compared with PSO and BFO summarization algorithms. To the best of the author's knowledge, this is the first research work for solving multi-document summarization using BSO algorithm. The objectives of the research work are as follows,

- The proposed summarization algorithm is used to produce optimal summary of the document
- Two DUC datasets is used to analyze the strength of summarization algorithms
- The performance comparison is analyzed using ROUGE score

The organization of this research paper is as follows; the related works are discussed in the section 2. Section 3 discusses about the conventional BSO. Section 4 discusses the proposed MDSBSO. The experimental results and discussions are given in section 5. Finally, conclusion of this research work is discussed in section 6.

2. Related works

The document summarization gains more attention among many researchers and developers to develop an efficient summarization model to fulfil the requirements of the end user. The nature inspired optimization algorithms plays a major responsibility for solving the document summarization problem. Hence, this section discusses some of the methods in the field of document summarization. Nandhini et al. (2014) designed an improved DE (IDE) algorithm for document summarization problem [27]. Ouyang et al. (2011) presents a regression model to make a query-focused multi-document summarization. The support vector regression (SVR) model is used to guess the significance of a sentence from given documents [5]. Fattah et al. (2009) designed a new content selection approach for automatic text summarization with two major phases. First, features are trained using GA and mathematical regression (MR) models to achieve an appropriate combination of feature weights. Then, the appropriate feature is considered as inputs to the Gaussian mixture model (GMM) in order to build an optimal text summarization [3].

Nandhini et al. (2016) developed an interactive GA-based individualized summarization to exploit the readability of significant sentences [28]. Mirshojaee et al. (2020) developed a multi-agent meta-heuristic optimization algorithm (MAMHOA) for extractive text summarization [29]. The MAMHOA scheme is a combination of multi-agent systems and biogeography-based optimization (BBO) algorithm. Rautray et al. (2019) developed a new cuckoo search-based multi-document summary extractor (CSMDSE) [30].

Yuan et al. (2020) designed an abstractive summarization method that combines word attenuation with multilayer convolutional neural networks (CNNs) to extend a standard sequence-to-sequence (seq2seq) model [31]. Patel et al. (2019) developed new multi-document summarization algorithm to expand good content exposure with information diversity [32]. A statistical feature based technique that exploits the fuzzy technique that dealt with the uncertainty and imprecise of feature weight. In addition, cosine similarity used to remove redundant information from the given document to improve the performance. Rautray et al. (2015) developed a new population-based stochastic optimization based summarization for comparisons study to solve document summarization problem. It identifies the relationship between sentences based on similarity and reduces the weight of each sentence to remove summary sentences at different compression stage. A comparison of both the optimization methods based on the fallout value of extracting sentences demonstrates the good performance of PSO in contrast with DE on five English corpus data [9].

Algorithm 1: Conventional BSO
1: Initialization phase
Step 1.1: Randomly initialize n ideas and required parameters
2: Clustering phase
Step 2.1: Cluster n idea into m cluster using clustering algorithm
Step 2.2: Assign the ranking values for each cluster and record the best individual idea as cluster center in each cluster
Step 2.2: If ($rand() < P_{replace}$)
Randomly choose the cluster center
Randomly generate an idea to replace chosen cluster center
End
3: Generation phase
Step 3.1: For i=1 to N
If ($rand() < P_{one}$)
Randomly choose the cluster center
If ($rand() < P_{one\_center}$)
Add the random values to the chosen cluster center in order to generate a new idea x_{new}

Else

Add random value to a random idea of the chosen cluster center to generate a new idea x_{new}
End if
Else
Randomly choose two cluster center
If $rand() < P_{two\_center}$
Combine the two cluster center and add random value to generate a new idea x_{new}
Else
Combine two random ideas from two clusters and added with random values to generate a new idea x_{new}
End if
End if
End for
4. Selection phase
Step 4.1: Newly generated ideas are compared with existing ideas then better ideas are stored as a new ideas
Step 4.2: If new ideas have been generated, go to step 3.1, otherwise go to step 4.3.
Step 4.3: If the termination condition is not satisfied then go to step 3, otherwise

3. Brainstorm optimization algorithm (BSO)

BSO algorithm is a well-known population-based swarm intelligence algorithm inspired by the behaviour of human brainstorm [18]. The brainstorm process helps common people to come up with diverse ideas. The good ideas are picked up from the groups of better diverged ideas. In the BSO algorithm, there are four major phases such as initialization, clustering, generation and selection. The description of conventional BSO algorithm is shown in Algorithm 1.

3.1 Initialization phase

In the initialization phase, the population is randomly generated with N ideas ($X_i = [x_{i1}, x_{i2}, ..., x_{iD}]$), where $1 < i < N$, N - is the population size and D is the problem size in the search space. Along with this, necessary parameters are also initialized at this stage.

3.2 Clustering phase

The clustering phase is used to generate the diverse ideas for speeding up the ability of searching process. In the BSO, the solutions are separated into several clusters. The clustering process is supported to pick up the good ideas and finds an optimal solution. The k-means clustering algorithm is used to find the cluster center of each cluster corresponds to the ideas, which are considered as optimum ideas among the given populations. In each clustering, the best ideas are recorded as cluster center based on the given threshold values. The probability value $P_{replace}$ employed to control the probability of replacing a cluster center by a randomly generated solution.

3.3 Generation phase

The new individual idea generation is used to achieve the global minimum for given solutions. For idea generation by piggyback, the new ideas generation is done with the help of old individual. It is written as

$$x_{new}^{i} = x_{old}^{i} + \xi(t) \times rand(t) \quad (1)$$

Where, x_{new}^{i} - is next new generations of the i^{th} idea. x_{old}^{i} - is the present i^{th} idea. $\xi(t)$ - is a coefficient values to the new idea.

$$x_{old}^{i} = w_1 * x_{old1}^{i} + w_2 * x_{old2}^{i} \quad (2)$$

Where, x_{old}^{i} is the value of the weighted summation of the i^{th} dimension of x_{old1} and x_{old2}. w_1 and w_2 are the weights coefficient to the contributions of two existing individuals. The coefficient of $\xi(t)$ is a weight contribution of randomly generated values to the new individual. It is written as follows,

$$\xi(t) = rand \times \log sig\left(\frac{0.5 \times Max\_Iter - Current\_Iter}{k}\right) \quad (3)$$

Where, $\log sig()$ is a logarithmic sigmoid transfer function. Max_Iter is the maximum number of iteration. $Current\_Iter$ is current iteration. k is a slope changing value of $\log sig()$.

Algorithm 2: Proposed MDSBSO
Step 1: Collect the set of multiple documents
<u>1. Pre-processing phase</u>
Step 1.1: Sentence segmentations
Step 1.2: Tokenization
Step 1.3: Removing stop word
Step 3.4: Stemming
<u>2. Input representation</u>
Step 2.1: Calculate the sentence informative score
Step 2.2: Calculate the similarity
Step 2.3: Choose the least similar sentences
Step 2.4: Merge all selected sentences
<u>3. Summary representations</u>
3.1: Initialization phase
Step 3.1.1: Randomly initialize n ideas and required parameters
3.2: Clustering phase
Step 3.2.1: Cluster n idea into m cluster
Step 3.2.2: Assign the ranking values for each cluster and record the best individual idea as cluster center in each cluster
Step 3.2.3: If ($rand() < P_{replace}$)
Randomly choose the cluster center
Randomly generate an idea to replace chosen cluster center
End if
3.3: Generation phase
Step 3.3.1: For i=1 to N

If ($rand() < P_{one}$)
  Randomly choose the cluster center
If ($rand() < P_{one\_center}$)
  Add the random values to the chosen cluster
  center to generate a new idea x_{new}
Else
  Add random value to a random idea of the chosen
  cluster center to generate a new idea x_{new}
End if
Else
  Randomly choose two cluster center
  If $rand() < P_{two\_center}$
    Combine the two cluster center and add random
    value to generate a new idea x_{new}
  Else
    Combine two random ideas from two clusters
    and added with random values to generate a
    new
    idea x_{new}
  End if
  End if
 End for

3.4 Selection phase

Selection of better idea is the most important task to evaluate the next iteration. In this phase, the cluster center is randomly chosen as optimal value. This phase will not simply perform in all iterations. However, it will perform when the probability value is small.

4. Proposed multi-document summarization using BSO (MDSBSO)

The BSO algorithm is proposed for multi-document summarization problem and the overview of proposed system is shown in Figure -1. The proposed MDSBSO is categorized into four phases including pre-processing phase, input representation phase, summary representation phase and summary selection phase.

4.1 Pre-processing phase

- **Sentence Segmentation:** Each individual document is denoted as D is segmented as $D = \{S_1, S_2......S_N\}$. S_j denotes the j^{th} sentence in the document. N is the number of sentences in the document.
- **Tokenization:** The sentences are tokenized as $T = \{t_1, t_2......t_m\}$ for t_k is $k = 1, 2, ...m$, m is number of tokens/terms.
- **Removing stop word:** Less significance words are removed with respect to the document. For instances, ‘a’, ‘an’, and ‘the’ are low significant words in the English language.
- **Stemming:** Stemming method is used to remove the ends of words to common base form.

4.2 Input representation phase

The word form of pre-processed data is used to compute the weights for each sentence which is called a sentence informative score. The sentence informative score is calculated as follows,

$$tf_{ij} = \frac{freq_{ij}}{\max_i \ freq_{lj}} \qquad (4)$$

Here, $freq_{ij}$ -represent the number of occurrence of i^{th} word in j^{th} sentence. $freq_{lj}$ is represent l^{th} word in j^{th} sentence. $\max_i freq_{lj}$ -represents the maximum number of i^{th} word occurrence in j^{th} sentence. The weights of each word is calculated as follows,

$$w_{ij} = tf_{ij} * idf_{ij} \quad (5)$$

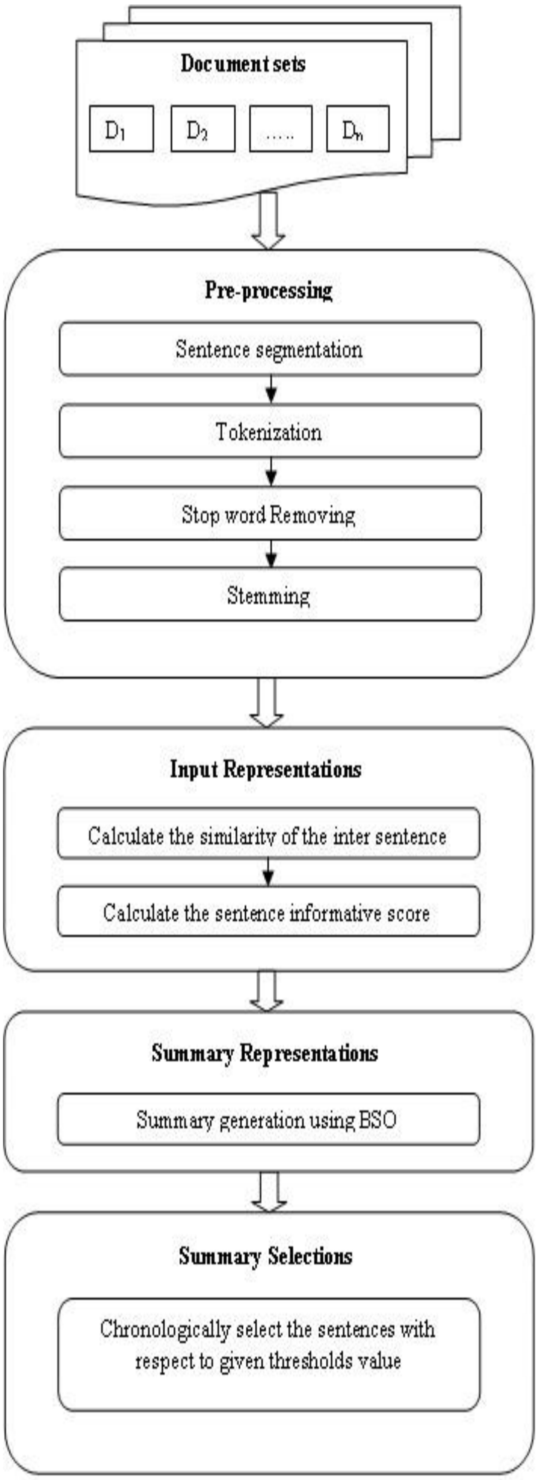


Figure 1 : Overview of proposed multi-document summarization

4. Selection phase

Step 4.1: Newly generated ideas are compared with existing ideas and then better ideas are stored as new ideas

Step 4.2: If new ideas are generated, go to step 4.3,otherwise go to step 3.3.

Step 4.3: If the termination condition is not satisfied then go to step 3.2, otherwise terminate the process
Step 4.4: Chronologically select the sentences with respect to given thresholds value.

Here, $idf_{ij} = \log \frac{N}{n_i}$, in which, N – is the number of sentence in the input text and n - is the number of sentences in each document. The similarity matrix is calculated as follows,

$$sim(s_i, q) = \frac{\sum_{i=1}^{t} w_{ij} * w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 * \sum_{i=1}^{t} w_{iq}^2}} \qquad (6)$$

Here, w_{ij} and w_{iq} represents the title input text weight and the weight of each word in document respectively. The similarity matrix is the comparison of sentence based on their keywords and essential words.

4.3 Summary representation phase

The aim of the summary representation phase is extraction of small set of useful information from the given documents. The optimal sentence selection process is performed by BSO algorithm using the sentence informative score based on the threshold value. Algorithm 2 shows the proposed MDSBSO.

4.4 **Summary selection phase**

In this phase, the optimal sentences are selected based on the given threshold value.

5. Experimental results and discussions

The performance of proposed MDSBSO document summarization algorithm is compared with PSO [36] and BFO [14]. The performance measures are calculated using ROUGE tool which is a well-known document summarization measuring tool [37]. The performance results are employed using MATLAB R2015 on windows 10 with Intel i3 and 4 GB RAM.

5.1 Datasets collections

Two benchmark datasets are used to analyze the performance of document summarization algorithms such as DUC 2006 and DUC 2007. The Table-1 shows the description about the datasets.

5.2 Parameter settings

Parameters setting of every nature inspired optimization algorithms are more significant to produce optimal results. An optimal parameter setting is shown in Table-2.

5.3 Performance measures

ROUGE is a well-known performance evaluation tool for document summarization problem to analyze the performance of the summarization algorithm. It is a software package that determines the similarity between human generated summary and machine generated summary. The high ROUGE score indicate highly informative summary and the low ROUGE score specify less informative summary. The ROUGE is defined based on various strategies including ROUGE-1, ROUGE-L, ROUGE-S, ROUGE-SU. ROUGE-1 used to asses overlap between the manual summary and the system summary. ROUGE-L calculates the ratio between the length of the longest common subsequence's (LCS) summary and the length of the reference summary. ROUGE-S used to asses overlap between ratio of the set of reference summaries and the candidate summary. ROUGE-SU is the advancement of ROUGE-S and added with unigram as the counting unit. The Precision (7), Recall (8) and F-Score (9) are the three criteria used to investigate the performance comparisons which are generated by ROUGE metric (Mirshojaee et al., 2020).

$$\Pr ecisions = \frac{\text{Re}\, levantSentence \cap RetrievedSentence}{RetrievedSentences} \quad (7)$$

$$\text{Re}\, call = \frac{\text{Re}\, levantSentences \cap RetrievedSentences}{RetrievedSentences} \quad (8)$$

$$F - Score = \frac{2 * precision * recall}{precision + recall} \quad (9)$$

5.4 Results analysis and discussions

The performance of the proposed MDSBSO summarization method obtains the best results when compared with PSO and BFO based summarization methods. Table 3 shows the experimental results of Precision, Recall, and F-Score using ROUGE-1. From the Table-3, it is evident that the proposed MDSBSO summarization algorithm produces higher enhancement when compared with PSO and BFO. According to ROUGE-L, the performance of the proposed MDSBSO summarization algorithm it produced slight enhancement when compared with PSO and BFO and performance results shown in Table-4. Table-5 shows performance results of Precisions, Recall, and F-Score using ROUGE-S. From the Table-5, it is evident that the proposed MDSBSO summarization algorithm produced higher accuracy when compared with PSO and BFO summarization algorithms.

Similarly, Table-6 demonstrates the performance of proposed MDSBSO summarization model using ROUGE-SU. Figure-2-4 demonstrates the performance comparisons of proposed MDSBSO summarization models on DUC 2006 datasets. Similarly, Figure 5-7 illustrates the performance comparisons of MDSBSO summarization model on DUC 2007 datasets. Hence, the experimental results confirmed that the proposed MDSBSO summarization method produced higher accuracy and optimal document summary.

Conclusions

Table 1 : Description about the datasets

Parameters of datasets	DUC 2006	DUC 2007
Number of groups	50	45
The number of documents (Each cluster)	25	25
Average	30.12	37.50
Maximum number of Sentences	79	125
Minimum number of Sentences	5	9
Summary length	250	250

Table 2 : Description about the parameters settings

S.No	PSO		BFO		MDSBSO	
	Parameters	Value	Parameters	Value	Parameters	Value
1.	P	50 doc	C	0.1	K	20
2.	C_1	0.2	P_{ed}	0.2	c	0.2
3.	C_2	0.2	N_c	200	P_{one_clus}	0.8
4.	$V_{\min}$	0.1	N_s	4	P_{one_center}	0.4
5.	$V_{\max}$	0.1	N_{re}	5	P_{two_center}	0.5
6.	W	0.45	N_{ed}	2	N	100
7.					M	5
8.					μ	0
9.					α	1
10.					$P_{replace}$	0.5

Table 3 : Performance results based on ROUGE-1

Methods	DUC-2006			DUC-2007		
	Precisions	**Recall**	**F-Score**	**Precisions**	**Recall**	**F-Score**
PSO	0.3725	0.4192	0.3944	0.2473	0.4303	0.3140
BFO	0.4591	0.4329	0.4456	0.2856	0.4195	0.3398
MDSBSO	0.5485	0.4495	0.4940	0.3174	0.4281	0.3645

Table 4 : Performance results based on ROUGE-L

Methods	DUC-2006			DUC-2007		
	Precisions	**Recall**	**F-Score**	**Precisions**	**Recall**	**F-Score**
PSO	0.1725	0.0902	0.1184	0.0938	0.0874	0.0904
BFO	0.1982	0.0969	0.1301	0.1172	0.0951	0.1050
MDSBSO	0.2185	0.1295	0.1626	0.1972	0.1836	0.1901

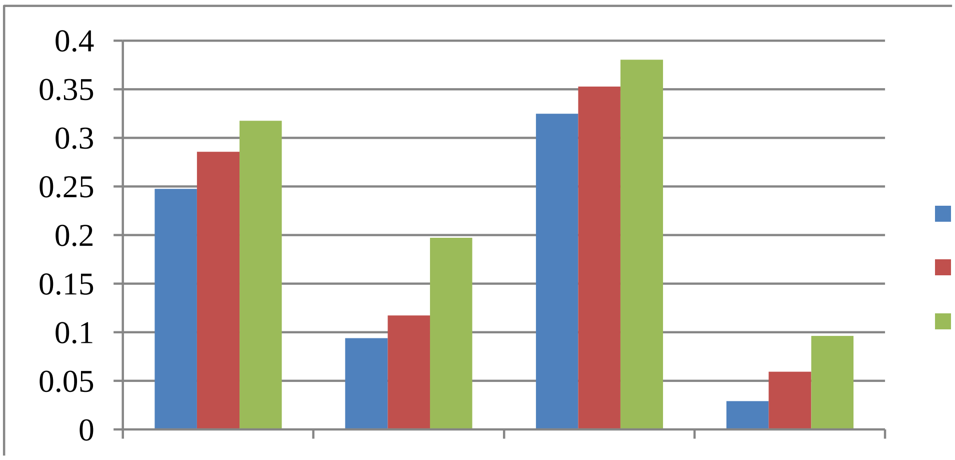
Table 5 : Performance results based on ROUGE-S

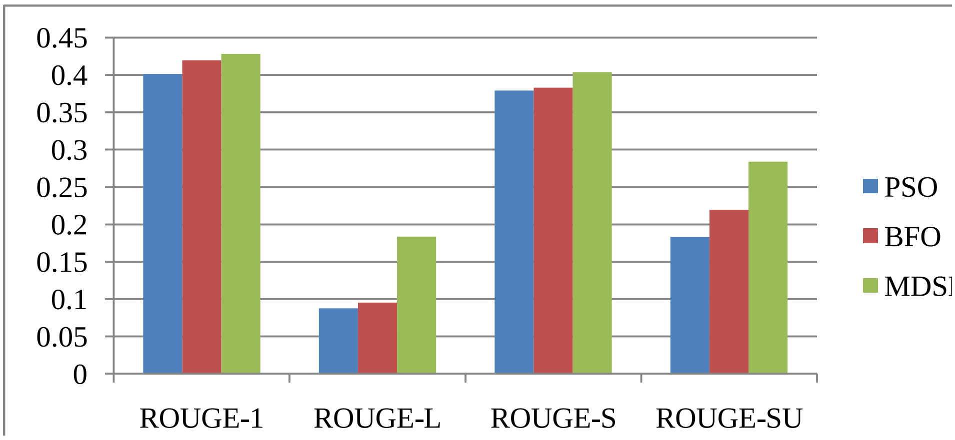
Methods	DUC-2006			DUC-2007		
	Precisions	**Recall**	**F-Score**	**Precisions**	**Recall**	**F-Score**
PSO	0.3728	0.4229	0.3962	0.3248	0.3791	0.3498
BFO	0.4028	0.4528	0.4263	0.3527	0.3831	0.3672
MDSBSO	0.4739	0.4890	0.4813	0.3802	0.4037	0.3916

Table 6 : Performance results based on ROUGE- SU

Methods	DUC-2006			DUC-2007		
	Precisions	**Recall**	**F-Score**	**Precisions**	**Recall**	**F-Score**
PSO	0.0462	0.2902	0.0797	0.0291	0.1830	0.0502
BFO	0.0832	0.3201	0.1320	0.0592	0.2195	0.0932
MDSBSO	0.1931	0.3691	0.2535	0.0961	0.2841	0.1436

In this research paper, the BSO algorithm is applied to multi-documents summarization to extract optimal summary (MDSBSO). The proposed MDSBSO is compared with PSO and BFO summarization algorithms. The performance of all conversed summarization algorithms assessed in terms of the different ROUGE score. From the experimental results, it is determined that the performance of proposed MDSBSO based summarizer produces significant outcomes better than the PSO and BFO based summarization algorithms.





References

[1] G. Hu, S. Zhou, J. Guan, and X. Hu, "Towards effective document clustering: A constrained K-means based approach," Information Processing & Management, vol. 44, no. 4, pp. 1397-1409, 2008.

Figure 5 : Performances comparison based on precision values for DUC 2007

Figure 6 : Performances comparison based on Recall values for DUC 2007

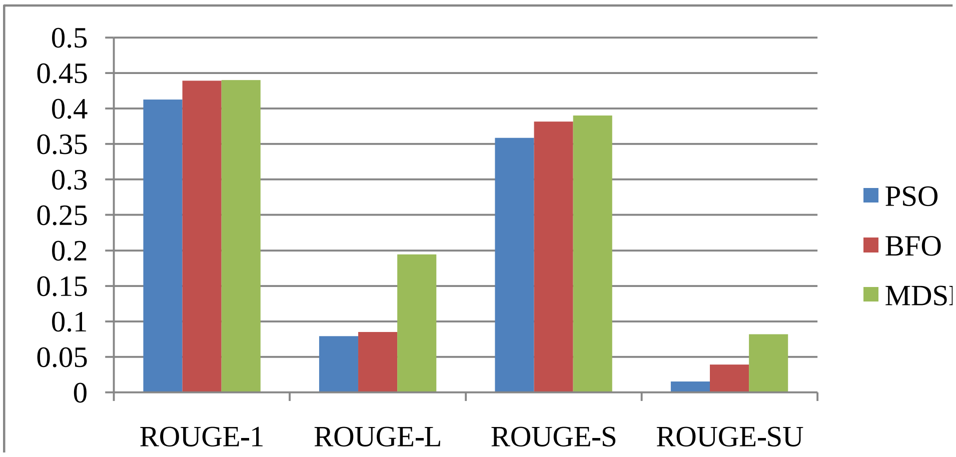


Figure 7 : Performances comparison based on F-Score values for DUC 2007

[2] M. A. Mosa, A. S. Anwar, and A. Hamouda, "A survey of multiple types of text summarization based on swarm intelligence optimization techniques," 2018.

[3] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," Computer Speech & Language, vol. 23, no. 1, pp. 126-144, 2009.

[4] R. M. Aliguliyev, "Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization," Computational Intelligence, vol. 26, no. 4, pp. 420-448, 2010.

[5] Y. Ouyang, W. Li, S. Li, and Q. Lu, "Applying regression models to query-focused multi-document summarization," Information Processing & Management, vol. 47, no. 2, pp. 227-237, 2011.

[6] M. Litvak, M. Last, and M. Friedman, "A new approach to improving multilingual summarization using a genetic algorithm," in Proceedings of the 48th annual meeting of the association for computational linguistics, 2010: Association for Computational Linguistics, pp. 927-936.
[7] R. M. Alguliev, R. M. Aliguliyev, and C. A. Mehdiyev, "Sentence selection for generic document summarization using an adaptive differential evolution algorithm," Swarm and Evolutionary Computation, vol. 1, no. 4, pp. 213-222, 2011.
[8] X. Cui, T. E. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," in Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005., 2005: IEEE, pp. 185-191.
[9] R. Rautray and R. C. Balabantaray, "Comparative study of DE and PSO over document summarization," in Intelligent Computing, Communication and Devices: Springer, 2015, pp. 371-377.
[10] O. F. HASSAN, "Text Summarization using Ant Colony Optimization Algorithm," Sudan University of Science and Technology, 2015.
[11] R. Rautray and R. C. Balabantaray, "CSTS: cuckoo search based model for text summarization," in Artificial Intelligence and Evolutionary Computations in Engineering Systems: Springer, 2017, pp. 141-150.
[12] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic text summarization using firefly algorithm," in 2019 Amity International Conference on Artificial Intelligence (AICAI), 2019: IEEE, pp. 61-65.
[13] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A combination of objective functions and hybrid Krill herd algorithm for text document clustering analysis," Engineering Applications of Artificial Intelligence, vol. 73, pp. 111-125, 2018.
[14] H. Asgari and B. Masoumi, "Provide a method to improve the performance of text summarization using bacterial foraging optimization algorithm," in the seventh iran data minig conference, 2013.
[15] T. R. Chandran, A. Reddy, and B. Janet, "An effective implementation of social spider optimization for text document clustering using single cluster approach," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018: IEEE, pp. 508-511.
[16] R. Rautray and R. C. Balabantaray, "Cat swarm optimization based evolutionary framework for multi document summarization," Physica a: statistical mechanics and its applications, vol. 477, pp. 174-186, 2017.
[17] D. Oliva and M. Abd Elaziz, "An improved brainstorm optimization using chaotic opposite-based learning with disruption operator for global optimization and feature selection," Soft Computing, pp. 1-22, 2020.
[18] Y. Shi, "An optimization algorithm based on brainstorming process," in Emerging Research on Swarm Intelligence and Algorithm Optimization: IGI Global, 2015, pp. 1-35.
[19] F. Pourpanah, Y. Shi, C. P. Lim, Q. Hao, and C. J. Tan, "Feature selection based on brain storm optimization for data classification," Applied Soft Computing, vol. 80, pp. 761-775, 2019.
[20] F. Pourpanah, C. P. Lim, X. Wang, C. J. Tan, M. Seera, and Y. Shi, "A hybrid model of fuzzy min–max and brain storm optimization for feature selection and data classification," Neurocomputing, vol. 333, pp. 440-451, 2019.
[21] Y. Shi, J. Xue, and Y. Wu, "Multi-objective optimization based on brain storm optimization algorithm," International Journal of Swarm Intelligence Research (IJSIR), vol. 4, no. 3, pp. 1-21, 2013.
[22] J. Chen, J. Wang, S. Cheng, and Y. Shi, "Brain storm optimization with agglomerative hierarchical clustering analysis," in International Conference on Swarm Intelligence, 2016: Springer, pp. 115-122.
[23] J. Liu, H. Peng, Z. Wu, J. Chen, and C. Deng, "Multi-strategy brain storm optimization algorithm with dynamic parameters adjustment," Applied Intelligence, pp. 1-27, 2020.
[24] R. A. Ibrahim, M. A. Elaziz, A. A. Ewees, I. M. Selim, and S. Lu, "Galaxy images classification using hybrid brain storm optimization with moth flame optimization," Journal of Astronomical Telescopes, Instruments, and Systems, vol. 4, no. 3, p. 038001, 2018.
[25] T. Zhang, C. Yang, and X. Zhao, "Using improved brainstorm optimization algorithm for hardware/software partitioning," Applied Sciences, vol. 9, no. 5, p. 866, 2019.
[26] X.-R. Chen, J.-Q. Li, Y. Han, B. Niu, L. Liu, and B. Zhang, "An Improved Brain Storm Optimization for a Hybrid Renewable Energy System," IEEE Access, vol. 7, pp. 49513-49526, 2019.
[27] K. Nandhini and S. R. Balasundaram, "Extracting easy to understand summary using differential evolution algorithm," Swarm and Evolutionary Computation, vol. 16, pp. 19-27, 2014.
[28] K. Nandhini and S. R. Balasundaram, "Improving readability through individualized summary extraction, using interactive genetic algorithm," Applied Artificial Intelligence, vol. 30, no. 7, pp. 635-661, 2016.
[29] S. H. Mirshojaee, B. Masoumi, and E. Zeinali, "MAMHOA: a multi-agent meta-heuristic optimization algorithm with an approach for document summarization issues," Journal of Ambient Intelligence and Humanized Computing, pp. 1-16, 2020.
[30] R. Rautray, R. C. Balabantaray, R. Dash, and R. Dash, "CSMDSE-Cuckoo Search Based Multi Document Summary Extractor: Cuckoo Search Based Summary Extractor," International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), vol. 13, no. 4, pp. 56-70, 2019.
[31] C. Yuan, Z. Bao, M. Sanderson, and Y. Tang, "Incorporating word attention with convolutional neural networks for abstractive summarization," World Wide Web, vol. 23, no. 1, pp. 267-287, 2020.

[32] D. Patel, S. Shah, and H. Chhinkaniwala, "Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique," Expert Systems with Applications, vol. 134, pp. 167-177, 2019.

[33] J. Tamilselvan and A. Senthilrajan, "Adding Text Document to Cluster Based on the Similarity Measures," International Journal of Pure and Applied Mathematics, vol. 118, no. 18, pp. 3069-3075, 2018.

[34] S. H. Mirshojaei and B. Masoomi, "Text summarization using cuckoo search optimization algorithm," Journal of Computer & Robotics, vol. 8, no. 2, pp. 19-24, 2015.

[35] S. Mandal, G. K. Singh, and A. Pal, "Text Summarization Technique by Sentiment Analysis and Cuckoo Search Algorithm," Singapore, 2020: Springer Singapore, in Computing in Engineering and Technology, pp. 357-366.

[36] R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic single-document text summarization using particle swarm optimization algorithm," Procedia Computer Science, vol. 117, pp. 30-37, 2017.

[37] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries ACL," in Proceedings of Workshop on Text Summarization Branches Out Post Conference Workshop of ACL, 2004, pp. 2017-05.