Karbala International Journal of Modern Science

Volume 9 | Issue 1 Article 6

SMATS: Single and Multi Automatic Text Summarization

Siba Prasad Patil
Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India

Rasmita Rautray
Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India, rasmitarautray@yahoo.co.in

Recommended Citation

Patil, Siba Prasad and Rautray, Rasmita (2023) "SMATS: Single and Multi Automatic Text Summarization," *Karbala International Journal of Modern Science*: Vol. 9 : Iss. 1 , Article 6.
Available at: https://doi.org/10.33640/2405-609X.3281

SMATS: Single and Multi Automatic Text Summarization

Abstract

In today's world, the rapid growth of textual data on internet sites & online resources makes it challenging for human being to assimilate essential information. To handle such issues, text summarization (TS) plays an important role. Through the TS process, a shorter version of the original content is generated to preserve the relevant information. This study suggests a quantitative assessment of models for single and multi-document summarization based on the sentence scoring method. Experimentation of the models has been carried out on DUC datasets. A detailed comparative analysis of the models is reported with respect to the performance of algorithms based on various metrics such as Recall Oriented-Understudy for Gisting Evaluation (ROUGE), Range, Co-efficient of Variation (CV) and Readability score.

Keywords

Extractive Text Summarization; Ant Colony Optimization; Bat Algorithm; Cuckoo Search Optimization; Firefly Algorithm; Flower Pollination Algorithm

RESEARCH PAPER

SMATS: Single and Multi Automatic Text Summarization

Siba P. Pati, Rasmita Rautray*

Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India

Abstract

In today's world, the rapid growth of textual data on internet sites & online resources makes it challenging for human being to assimilate essential information. To handle such issues, text summarization (TS) plays an important role. Through the TS process, a shorter version of the original content is generated to preserve the relevant information. This study suggests a quantitative assessment of models for single and multi-document summarization based on the sentence scoring method. Experimentation of the models has been carried out on DUC datasets. A detailed comparative analysis of the models is reported with respect to the performance of algorithms based on various metrics such as Recall Oriented-Understudy for Gisting Evaluation (ROUGE), Range, Co-efficient of Variation (CV) and Readability score.

Keywords: Extractive text summarization, Ant colony optimization, Bat algorithm, Cuckoo search optimization, Firefly algorithm, Flower pollination algorithm

1. Introduction

Currently, humans utilise the internet, social media, the World Wide Web etc. Extensively, which has accelerated the rate of information growth and led to information overload. It inspires the researchers to give their efforts on text summarization as a solution to this problem. It seeks to produce a concise or compressed version of the original text document without losing its core contents, known as summary. Usually, a summary saves the time to get the required data from the document for the users & also improves the readability of a document.

Typically, categorization of TS based approaches is of two types: extractive & abstractive [1]. An extractive technique recognizes & derives the most significant sentences from the original document in order to generate the summary. In extractive, no modification is done to the derived sentences while abstractive technique produces a concise summary using its innovative words. Through the use of different words, it conveys the ideas of the documents. In Natural language Processing (NLP), extractive is generally faster, more versatile, and more efficient than abstractive. However, abstractive is more proficient in comparison to extractive because it has the capability to produce human-comprehending summaries. TS can also be categorized as indicative & informative. Generally, an indicative summary highlights the theme of the input document for quick classification. But in an informative summary, these topics are defined along with conclusions, suggestions & recommendations based on user interests.

Based on source document length, TS is classified as: single document summarization (SDS) & multi document summarization (MDS). If the summary is produced from only one document at a time, then it's termed as SDS. However, MDS is to make a brief

Received 16 August 2022; revised 12 December 2022; accepted 14 December 2022.
Available online 13 January 2023

* Corresponding author.
E-mail address: rasmitarautray@yahoo.co.in (R. Rautray).

https://doi.org/10.33640/2405-609X.3281

summary from multiple numbers of documents. Theoretically, this is an extended concept of SDS.

1.1. Motivation

Utilizing information from various sources, including the World Wide Web, posts on social media, wikis, etc., effectively is the major goal of this study. The users' busy schedule & the needs in work & social sectors gives motivation to find a model which makes easy for the human beings to get the much-needed textual data from the original input document in the shortest period of time. Therefore, the automatic text summarization will be a superior option for effective information usage to meet user needs. In essence, it enables the user to pick out the content that is most important to them, saving them time and effort. A sophisticated & futuristic model is developed in this work to provide a high-quality summary of a document. Additionally, this model can be used in MDS and SDS.

1.2. Contribution

In order to extract the information efficiently, a model is designed for both SDS & MDS. Since the MDS has a larger search space than SDS, hence it makes quite difficult for the user to identify the most valuable sentences. Therefore, few nature-inspired optimization algorithms like Ant Colony Optimization (ACO), Bat Algorithm (BA), Cuckoo Search Optimization (CSO), Firefly Algorithm (FA) & Flower Pollination Algorithm (FPA) are implemented to fetch the most significant sentences from the source documents. Although, a lot of optimization algorithms have been used in this field, but in this paper, a few rarely used algorithms (BA & FPA) are also implemented to show their efficiency. The performance of these algorithms is estimated on the basis of ROUGE score, Range, Co-efficient of Variation (CV) & Readability metrics. The model presented in this paper has the ability to generate the summary that includes the whole content of the document without reducing its readability & verbal quality.

The remainder of this paper is structured as following: Section 2 highlights a detailed review of related work; Section 3 demonstrates methodology and the experimental results are discussed in Section 4 trailed by conclusion in Section 5.

2. Literature survey

A handful of the nature-inspired optimization techniques used in recent years to handle the summarization problem; a few of them are highlighted in Table 1.

The authors of [2] have been put forth a novel MDSCSA model for the purpose of multi-document summarization. In this paper, the model compared with PSOS and CSOS models and the outcomes were evaluated in terms of ROUGE score & it clearly showed that the model approached in the paper gave better results than the other summaries. In [3], a Multi-Document Temporal Summarization (MDTS) method was presented which is capable of generating a summary, depending upon the temporally related events derived from multiple documents. These were extracted along with time stamp using TIMEML standard tags. Its performance has been compared with PSOS, CSOS and MDSCSA & found MDTS performs better than these methods. An advanced CSMDSE model was proposed for multi-document summarization in [4]. This model has been compared with some other summary extractor techniques like CSOE, PSOE, IPSOE & ACOE and the experimental results show that this model outperformed other methods. A

Table 1. Use of optimization algorithms in text summarization.

Sl. No.	Authors (Year)	Applied Algorithms	Size of Source Document
1	Mamidala, Kishore Kumar, 2021 [3]	CSO	Multi
2	Tomer, Minakshi, 2021 [8]	FA	Multi
3	Krishnan, N., 2021 [12]	FPA	Single
4	Ali, Zuhair Hussein et al., 2019 [5]	CSO	Multi
5	Rautray, Rasmita, et al., 2019 [4]	CSO	Multi
6	Pattanaik, Anshuman, et al., 2019 [17]	BA	Single
7	Al-Abdallah, Raed Z., 2019 [9]	FA	Single
8	Rautray, Rasmita, 2018 [2]	CSO	Multi
9	Al-Saleh, Asma, 2018 [22]	ACO	Multi
10	Setyadi, I. Wayan Adi et al., 2018 [23]	ACO	Multi
11	Rautray, Rasmita, 2017 [6]	CSO	Single
12	Rautray, Rasmita, 2017 [7]	CSO	Multi
13	Ali, Zuhair Hussein, 2017 [10]	FA	Multi
14	Yang, Xin-She, 2013 [11]	FA	Multi

VIKOR algorithm-based CS model for multi-document summarization has been introduced in [5] and put into practice. The result of the model has evaluated using ROUGE metric & it demonstrates the model's effectiveness. In [6], few optimization algorithms including CS, CSO, PSO, HS and DE have been comparatively analysed for single document text summarization. To verify the relevance and exclusivity of the summary, output of each algorithm has evaluated using a variety of metrics, including F-score, recall, and precision values. It revealed that the CS algorithm required less parameter for tuning than other algorithms. Another model based on the CS algorithm named as CSTS has been also implemented for text summarization using cosine similarity in [7].

The Firefly Algorithm (FFA) has been used to explore a model for multiple document summarization based on noble swarm intelligence [8]. Using ROUGE metrics, subject relation factor, cohesiveness factor, and readability metrics, the performance of this algorithm has compared to PSO and Genetic Algorithm (GA), and the results showed that it performs noticeably better than the other two. However, FFA has also used and compared with two evolutionary algorithms, GA and HS in [9]. The proposed strategy has higher ROUGE score values than the other two ways, according to the evaluation of the results using ROUGE metrics and the EASC Corpus. In [10], the FFA has integrated as association rule mining to reduce the set of rules produced by the Fuzzy logic system. The effectiveness of this model has shown with the help of ROUGE value. The authors of [11] have explained some current applications of firefly algorithm & the advantages of its usage.

A text summarization method using FPA has proposed based on an ontology that performed exceptionally well on MS Marco data set in [12]. This technique has shown very efficient in terms of performance. The authors of [13] showed that it is possible to fix the fossil fuel exhaustion issue with the help of FPA. The authors applied MOPFA model in order to crack the multi-objective problems in [14]. And the result proves that it has more effective as compare to the MOEA model. The authors of [15] have implemented the FPA for selection of optimal bus by employing the PMU in order to ensure the highest level of bus observance throughout the system. A robot calibration strategy combining EKF and ANN built on the Butterfly & Flower Pollination Algorithm (abbreviated as ANN-BFPA) has been proposed in [16]. The robots precise pose development has done using this model.

A BA-based optimization approach with the primary objective of reducing sentence redundancy in the final generated summary has discussed in [17]. The outcomes demonstrated the effectiveness of the provided model. However, in [18], the authors have proposed a hybrid inversion method using BA to solve electromagnetic inverse scattering issues. To make optimization effective for a variety of optimization problems and to take advantage of the dynamic membrane computing framework in [19], the authors have suggested an upgraded BA called DMBA. Results were assessed based on their diversity and exploitation potential, which demonstrated the effectiveness of the method. In order to choose the best position and the number of voltage dip monitoring sensors, the Binary Bat Algorithm was also used in [20]. For the best placement and sizing of Distribution Static Compensators, the authors of [21] have been approached using a combined strategy of NVSI and BA (D-STATCOM).

The authors of [22] used an extractive summarizer built on ACO to extract the significant sentences from the input text document. ROUGE score has used to evaluate the outcomes and demonstrate the effectiveness of the method. However, in [23], authors have combined both Graph and ACO techniques to address and solve the summarization issues. The implementations of ACO technique have also used in the field of UAV (Unmanned Aerial Vehicle) based intelligent pesticide irrigation system and unmanned surface vessels for constraint path-planning control in [24,25]. For the goal of joint virtual function placement & routing, the ACO has been integrated with fuzzy heuristic (referred to as FH-ACO) in [26]. Additionally, the ACO has used to the evolution obstacle for software project scheduling and task offloading in fog computing in [27,28].

3. Methodology

The proposed SMATS model is primarily designed using extractive TS for both single document as well as multiple documents. Fig. 1 demonstrates the overview of this model. This model includes five steps, such as: text pre-processing, sentence score Evaluation, sentence similarity evaluation, cuckoo search implementation & summary generation.

3.1. Text pre-processing

In pre-processing, DUC-2003 dataset is used for SDS and DUC-2005 dataset is used in case of MDS. The pre-processing steps are illustrated in Fig. 2.

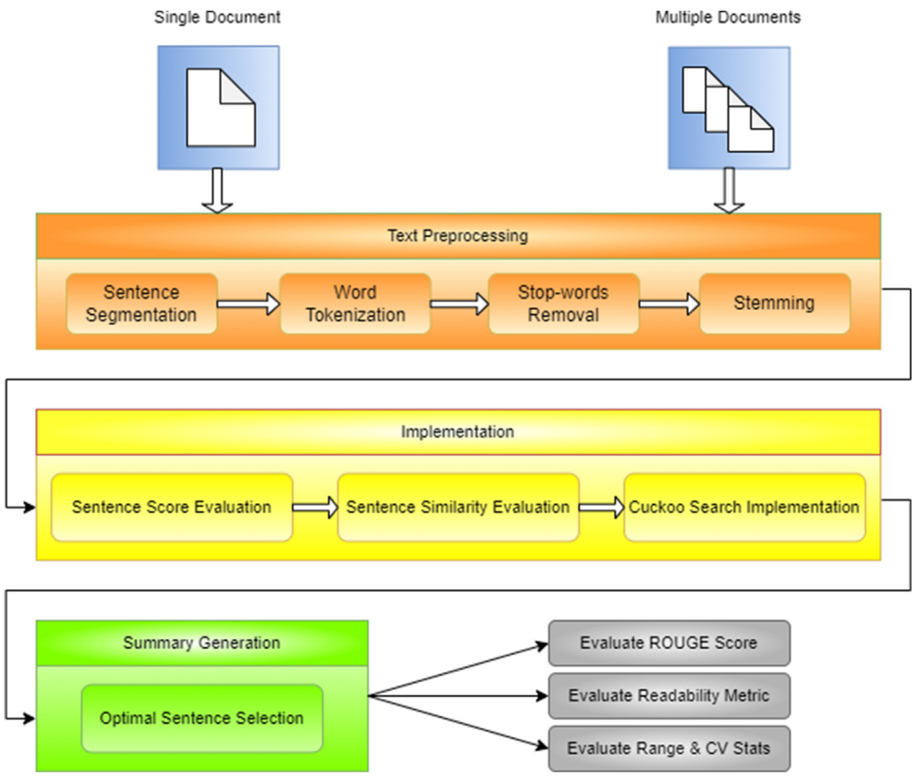

Fig. 1. SMATS model.

3.1.1. Sentence segmentation

The way of extracting & separating sentences from a document individually is known as Sentence Segmentation. Mostly, the punctuation, period character or full stops are applied to separate the sentences.

3.1.2. Word tokenization

It is defined as the way of dividing the words from each sentence into tokens. White space is usually used to separate these tokens.

3.1.3. Stop-words removal

Stop-words are often used terms which have less importance in a text document, such as "a," "an," "the," "'s," etc. In specific circumstances, certain phrases can also be treated as stop words. In this step, the terms of this nature are eliminated.

3.1.4. Stemming

It is referred to as the process of reverting subsequent words to their original or root form.

3.2. Sentence score evaluation

It includes the calculation of the weightage value of each sentence. For single document, Term Frequency (TF) is computed for each term in the text document that shows the incidence of the terms. However, for multi-document, Term Frequency Inverse Document Frequency (TF-IDF) values are evaluated of all the terms of all the documents in the dataset. Then the sentence score is measured for each sentence with the help of summation of TF values for single document and TF-IDF values for multi-document.

3.3. Sentence similarity evaluation

In this step, cosine similarity method is employed to calculate the sentence similarity scores for all the sentences using the sentence score values measured in the previous step. These similarity values are kept in a sentence similarity matrix which will be further applied as input to CSO algorithm. Equation (1) demonstrates the formula which is used to evaluate the cosine similarity values.

$$\text{Cosine}(S_i, S_j) = \frac{\sum_{k=1}^{m} S_{ik} S_{jk}}{\sqrt{\sum_{k=1}^{m} S_{ik}^2 \cdot \sum_{k=1}^{m} S_{jk}^2}} \qquad (1)$$

3.4. Cuckoo search implementation

Currently, the CSO is a latest nature-inspired optimization algorithm, motivated by a particular

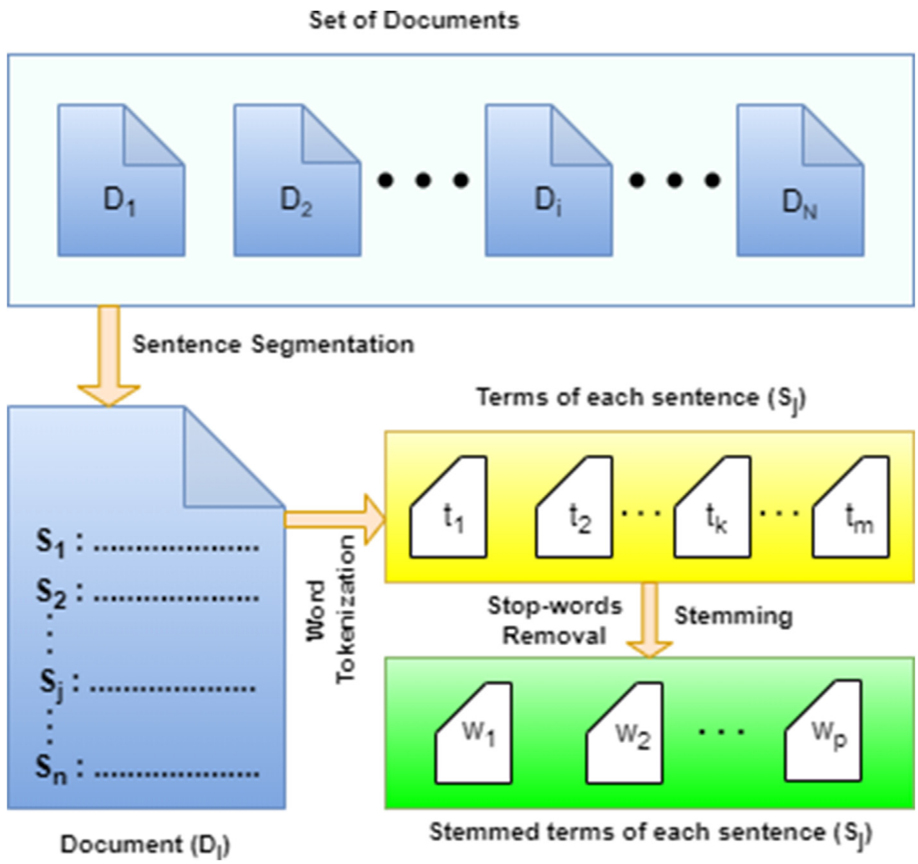

Fig. 2. Steps of pre-processing.

species of bird called cuckoo. Due to the aggressive reproduction nature & sweet sounds, these birds are so mesmerized. The mature cuckoos put their eggs into other host bird's nests. Each nest contains an egg, which denotes a solution & every cuckoo can only lay a single egg that demonstrates a latest & potentially better solution. Generally, in CSO algorithm, each nest has only one egg where there is not a difference between eggs, nest or cuckoo because single individual nest denotes one egg and it also signifies one cuckoo. In addition, it is possible to extend this algorithm to each nest with multiple eggs showing a solution set. There are three idealized rules which can describe the basic fundamental concept of CSO algorithm.

- Each cuckoo places one egg in a random nest as a solution set.
- The next generation will be the nest having the best eggs.
- A fixed number of nests are available and a host bird has a probability $P_a \in (0,1)$ of discovering an alien egg. If it occurs, the host can discard the egg or abandon the nest & build a new one elsewhere.

The global explorative random walk is combined with a local random walk to generate new solutions x_i^{t+1}, which can be constrained by the switch probability P_a. The local random walk can be defined by Equation (2).

$$x_i^{t+1} = x_i^t + \propto \times S \otimes H(P_a - \varepsilon) \otimes \left(x_j^t - x_k^t\right) \tag{2}$$

where x_j^t & x_k^t are distinct solutions picked by random permutation

$H(u)$ refers to Heaviside function

ϵ denotes a no. picked randomly from a uniform distribution

S is size of the step

Meanwhile, the global random walk is done with the help of Levy flights which consists of a consecutive random steps & is considered as a series of rapid jumps. The global walk can be written as (Equation (3)):

$$x_i^{t+1} = x_i^t + \propto \otimes Levy(\lambda) \tag{3}$$

where α is step size that must be proportionate to optimization problem scale (i.e., $\alpha > 0$)

$\otimes$ refers to entry-wise move through multiplication

$Levy(\lambda)$ are random no. picked from Levy distribution

3.4.1. CS algorithm parameter setting

Parameter setting in implementation of optimization algorithms has a major role to solve any specific problem. Hence, suitable parameters should be selected for the improvement in the accuracy of the result & the system performance. In this problem, the various parameters applied in the CS algorithm are provided in Table 2.

The steps included in the algorithm (represented in Fig. 3) for implementation in TS of both sizes of source documents are discussed below.

Step 1: Collect the pre-processed documents w_1, $w_2, \ldots w_p$. Each document's length is determined by the no. of sentences present in it which may differ document-wise.

Step 2: Compute the sentence score S_{jk} for each sentence S_j of the pre-processed documents using Equations (4) and (5).

$$S_{jk} = tf_{jk} \tag{4}$$

$$S_{ji} = tf_{ik} \times \log\left(n / df_k\right) \tag{5}$$

where S_{jk} is Sentence score for single document

S_{ji} is Sentence score for multi-documents

tf_{jk} refers to Term Frequency i.e., occurrence of t_k in sentence S_j

tf_{ik} refers to Term Frequency of t_k in document D_i

n denotes the no. of documents

df_k represents Document Frequency of t_k

$\log(n / df_k)$ refers to Inverse Document Frequency.

Step 3: Evaluate sentence similarity score using cosine similarity metric for each document using Equation (1).

Step 4: For single document, fetch the least similar sentences with respect to a threshold value. But in case of multi document, create one document containing all the least similar sentences from all the documents.

Step 5: Set the initial parameters such as population size, rate of alien eggs, step size & levy flight.

Table 2. CS parameter configuration.

Parameters	Values
Rate of alien eggs	0.75
Step factor	0.5
Levy exponent	0.8
Population size	Varies for different document

Step 6: The sentence similarity scores of these selected sentences are stored in a matrix which is taken as an input to CSO algorithm as each cuckoo's nest information.

Step 7: Calculate fitness values for all the nests using Equations (4) and (5).

Step 8: Generate the new population of nests with the help of Levy flight as provided in Equation (3).

Step 9: Compute the fitness values f_{new} of new nests & validate with f_{prev} of previous nests.

Step 10: If f_{new} is better than f_{prev}, then change f_{prev} by f_{new}.

Step 11: Select the probability P_a of the worst performing nests in the new population. In the given search space, replace them with random generated ones and create new ones.

Step 12: Evaluate fitness function of newly generated nests.

Step 13: Determine the best nests in the current population based upon fitness values. A comparison is then made between these nests & the best nest obtained so far and the current best is replaced by the previous best.

Step 14: Go to Step 7 till the termination criterion is encountered. The final step of this algorithm is elaborated in Section 3.5.

3.5. Summary generation

The ultimate summaries are generated after deriving the sentences sequentially from the input document for both single document and multi-document. After summary generation, a few result evaluation metrics are estimated to compare the summaries generated by the optimization algorithms.

3.5.1. Fitness function

A summary can be easily readable by the user if its contents have covered all the topics of the original document and its sentences should be strongly related to each other. Therefore, two factors such as: cohesion & readability have been used to formulate the fitness function (F) to make the summary more informative. The function used to extract the summary is presented in Equation (6).

$$F = F(CF) + F(RF) \tag{6}$$

$$F(CF) = 1 - Sim\left(S_i, S_j\right), \quad i \neq j = 1, 2, \ldots, n$$

$$F(RF) = Sim\left(S_i, S_j\right), \quad i \neq j = 1, 2, \ldots, n$$

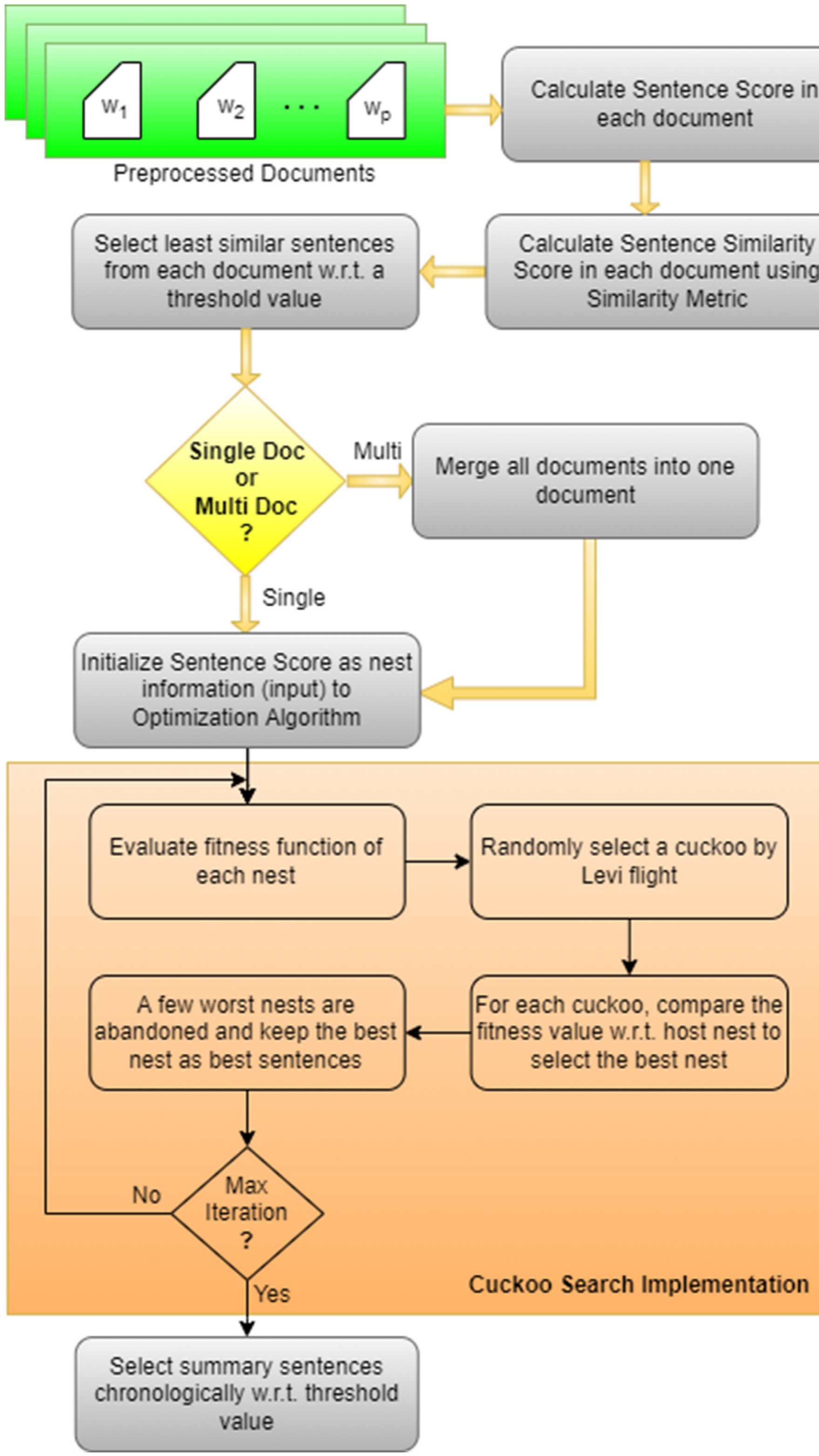

Fig. 3. Flowchart of CSO Implementation.

where $F(CF)$ refers to the Cohesion Factor (CF) which determines whether the summary sentences are related to original topic or not.

$F(RF)$ is the Readability Factor (RF) which shows the smoothness of the summary & the similarity between the summary sentences.

$Sim(S_i, S_j)$ measure the similarity between sentences S_i and S_j.

4. Experiment & result analysis

In this section, the performance analysis of the optimization algorithms is illustrated for both single document & multi-document over DUC-2003 dataset & DUC-2005 dataset respectively. All the experimental work is done on Jupiter Notebook Anaconda3 (Version- 2020.11) in Windows 10 64-bit operating system with Intel Core i5 CPU @1.60 GHz, 1.80 GHz processor (8 GB RAM). The python version is Python 3.8.5.

A threshold value is assigned for each algorithm to create the final summaries of both single-document & multi-document. These summaries are termed as system summaries. For a comparison of these system summaries, the results are estimated in the form of ROUGE score, Range, Co-efficient of Variation (CV) Statistics & Readability metrics. ROUGE-1 evaluates the unigrams overlapping between each system summary & reference summary, whereas ROUGE-2 computes overlapping of bigrams. ROUGE-L recognizes the longest common sub-sequence of n-grams automatically.

Additionally, there are two more factors; Range & CV, which give better insight into the algorithm's efficiency. The range is known as the difference between ROUGE-Best & ROUGE-Worst & illustrated in Equation (7) whereas CV is defined by the Equation (8). But, Readability metrics of a summary mean "whether the generated summary will be easily readable or not". It also refers to the understanding of the summary. There are also several metrics of readability score such as: Flesch Kincaid Grade Level (FKGL), Gunning fog Score (FOG), SMOG Index (SMOG), Coleman Liau (CL), Automated Readability Index (ARI).

$$Range = Rouge_{best} - Rouge_{worst} \tag{7}$$

$$CV = \frac{Range}{Rouge_{average}} \times 100 \tag{8}$$

4.1. Dataset

The DUC datasets are utilized to assess the result of the text documents. The descriptions of the dataset are demonstrated in Table 3 which is used in the current experimental work.

4.2. Performance analysis

In this section, the performance of all the implemented algorithms is compared for both single document as well as multi document summarization over DUC-2003 & DUC-2005 dataset respectively.

4.2.1. Single document summarization (SDS)

In case of SDS, the ROUGE score values of system summaries are computed with respect to a Reference summary. Table 4 shows the comparison of ROUGE-1 score values among system summaries while Table 5 & Table 6 demonstrates the comparison of ROUGE-2 & ROUGE-L values respectively. Then in Table 7, the Range scores of all optimization techniques are evaluated & the CV values of all the methods are computed in Table 8. Then the above discussed readability metrics are evaluated for each summary as represented in Fig. 4. Table 9 focuses on the performance comparison of cuckoo search-based model (CS-SMATS) with other existing models over the DUC-2003 dataset & it is graphically represented in Fig. 5.

After analysing the Tables 7–9, it is found that the summary created by CS-SMATS algorithm gives higher Range values, lower CV values & higher ROUGE score respectively which proves the efficiency of this model as compared to other models.

Table 3. Dataset description.

Dataset description	Single Document	Multi-Document
DUC Dataset	2003	2005
Sets of documents	5	25
Avg. sentences per doc.	4.20	8.41
Data source	TREC	TREC
Length of summary (in words)	120.40	244.15

Table 4. Evaluation of ROUGE-1 score of all algorithms for SDS.

	ACO	BA	FA	FPA	CS-SMATS
Doc-1	0.538	0.350	0.345	0.282	0.556
Doc-2	0.345	0.352	0.568	0.549	0.784
Doc-3	0.537	0.550	0.348	0.526	0.823
Doc-4	0.350	0.298	0.323	0.354	0.650
Doc-5	0.299	0.306	0.309	0.314	0.614

Table 5. Evaluation of ROUGE-2 score of all algorithms for SDS.

	ACO	BA	FA	FPA	CS-SMATS
Doc-1	0.499	0.259	0.259	0.249	0.499
Doc-2	0.249	0.249	0.499	0.509	0.749
Doc-3	0.499	0.506	0.265	0.499	0.795
Doc-4	0.215	0.221	0.218	0.225	0.612
Doc-5	0.249	0.249	0.249	0.249	0.556

Table 6. Evaluation of ROUGE-L score of all algorithms for SDS.

	ACO	BA	FA	FPA	CS-SMATS
Doc-1	0.531	0.327	0.335	0.282	0.544
Doc-2	0.325	0.305	0.544	0.538	0.767
Doc-3	0.529	0.528	0.333	0.526	0.808
Doc-4	0.312	0.298	0.294	0.326	0.650
Doc-5	0.291	0.291	0.299	0.299	0.608

Table 7. Analysis of range score of CS-SMATS algorithm with other algorithms for SDS.

		ACO	BA	FA	FPA	CS-SMATS
ROUGE-1	Best	0.538	0.550	0.568	0.549	0.823
	Worst	0.299	0.298	0.309	0.282	0.556
	Range	0.239	0.252	0.259	0.267	0.267
ROUGE-2	Best	0.499	0.506	0.499	0.509	0.795
	Worst	0.215	0.221	0.218	0.225	0.499
	Range	0.284	0.285	0.281	0.284	0.296
ROUGE-L	Best	0.531	0.528	0.544	0.538	0.808
	Worst	0.291	0.291	0.294	0.282	0.544
	Range	0.240	0.237	0.250	0.256	0.264

Table 8. Analysis of CV statistics of CS-SMATS algorithm with other algorithms for SDS.

		ACO	BA	FA	FPA	CS-SMATS
ROUGE-1	Range	0.239	0.252	0.259	0.267	0.267
	Average	0.413	0.371	0.378	0.405	0.685
	CV	57.86	67.92	68.51	65.92	38.97
ROUGE-2	Range	0.284	0.285	0.281	0.284	0.296
	Average	0.342	0.296	0.298	0.346	0.642
	CV	83.04	96.28	94.29	82.08	46.10
ROUGE-L	Range	0.240	0.237	0.250	0.256	0.264
	Average	0.397	0.349	0.361	0.394	0.675
	CV	60.45	67.90	69.25	64.97	39.11

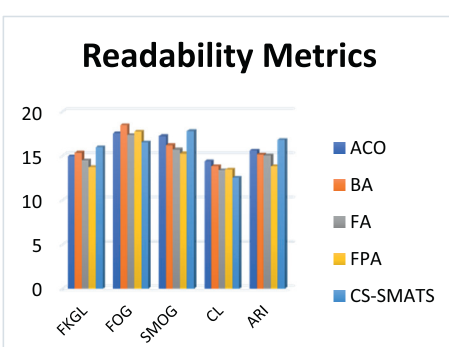

Fig. 4. Readability metrics comparison among optimization algorithms for single document.

Table 9. Performance comparison of CS-SMATS model with other existing models over DUC-2003.

Methods	Source	Evaluation Metric	
		ROUGE-1	ROUGE-2
LexRank	Peyrad M [29]	0.357	0.079
ICSI	Peyrad M [29]	0.376	0.094
FbTS	Tomer M [8]	0.441	0.160
CS-SMATS		0.685	0.642

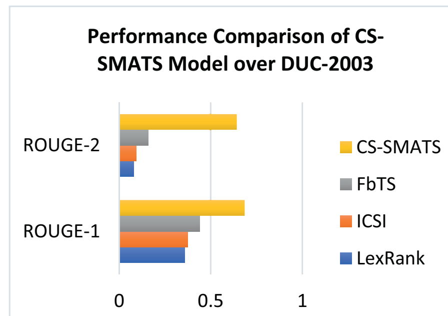

Fig. 5. Performance comparison of CS-SMATS model with other existing models over DUC-2003.

Table 10. Analysis of ROUGE scores of all algorithms for MDS.

		ACO	BA	FA	FPA	CS-SMATS
ROUGE-1	Best	0.566	0.471	0.487	0.550	0.684
	Worst	0.352	0.308	0.316	0.341	0.446
	Average	0.488	0.396	0.402	0.463	0.615
ROUGE-2	Best	0.501	0.419	0.434	0.490	0.586
	Worst	0.289	0.244	0.235	0.288	0.362
	Average	0.407	0.336	0.342	0.398	0.508
ROUGE-L	Best	0.551	0.468	0.479	0.532	0.640
	Worst	0.348	0.295	0.302	0.335	0.411
	Average	0.465	0.374	0.388	0.451	0.608

4.2.2. Multi document summarization (MDS)

In MDS, ROUGE, Range, CV Statistics & Readability metrics values of system summaries are computed. Table 10 demonstrates the comparison of all the ROUGE values. However, Tables 11 and 12 highlight the Range scores & CV values of all model

Table 11. Analysis of range score of CS-SMATS algorithm with other algorithms for MDS.

		ACO	BA	FA	FPA	CS-SMATS
ROUGE-1	Best	0.566	0.471	0.487	0.550	0.684
	Worst	0.352	0.308	0.316	0.341	0.446
	Range	0.214	0.163	0.171	0.209	0.238
ROUGE-2	Best	0.501	0.419	0.434	0.490	0.586
	Worst	0.289	0.244	0.235	0.288	0.362
	Range	0.212	0.175	0.199	0.202	0.224
ROUGE-L	Best	0.551	0.468	0.479	0.532	0.640
	Worst	0.348	0.295	0.302	0.335	0.411
	Range	0.203	0.173	0.177	0.197	0.229

Table 12. Analysis of CV statistics of CS-SMATS algorithm with other algorithms for MDS.

		ACO	BA	FA	FPA	CS-SMATS
ROUGE-1	Range	0.214	0.163	0.171	0.209	0.238
	Average	0.488	0.396	0.402	0.463	0.615
	CV	43.85	41.16	42.53	45.14	38.69
ROUGE-2	Range	0.212	0.175	0.199	0.202	0.224
	Average	0.407	0.336	0.342	0.398	0.508
	CV	52.08	52.08	58.18	50.75	44.09
ROUGE-L	Range	0.203	0.173	0.177	0.197	0.229
	Average	0.465	0.374	0.388	0.451	0.608
	CV	43.65	46.25	45.61	43.68	37.66

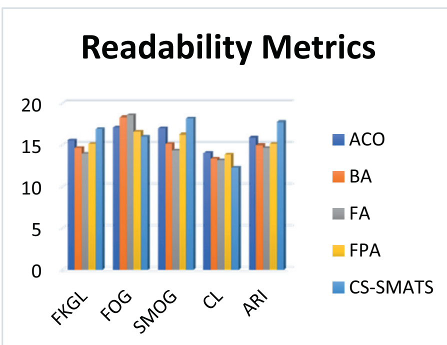

Fig. 6. Readability metrics comparison among optimization algorithms for multi document.

Table 13. Performance comparison of CS-SMATS model with other existing models over DUC-2005.

Methods	Source	Evaluation Metric	
		ROUGE-1	ROUGE-2
DESAMC + D	Alguliev R [30]	0.393	0.082
MMR	Alguliev R [31]	0.347	0.060
PLSA	Alguliev R [30]	0.391	0.081
MCMR	Alguliev R [31]	0.389	0.079
SVR	Alguliev R [31]	0.384	0.075
CS-SMATS		0.615	0.508

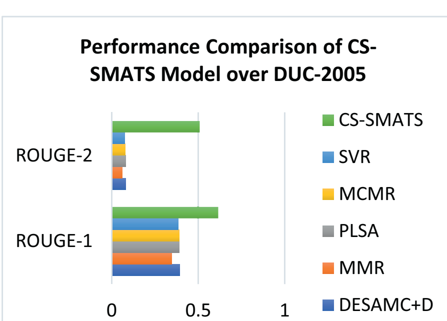

Fig. 7. Performance comparison of CS-SMATS model with other existing models over DUC-2005.

summaries respectively. Fig. 6 demonstrates the graphical analysis of Readability metrics of all model summaries. Table 13 focuses on the performance comparison of CS-SMATS model with other existing models over the DUC-2005 dataset & it is graphically represented in Fig. 7. For multi document summarization, it's clearly visible that CS-SMATS algorithm also results significantly better than other models based on Range, CV statistics & Readability metrics.

5. Conclusion

This paper is focused on a single and multi-automatic text summarization model which is inspired by the cuckoo search optimization algorithm. This model aims to produce a brief or compressed text document from the input document without losing its principle, termed as summary. It reduces the users' time to fetch the required data from the document & also makes easy to take decision whether a document is readable or not. The major contributions of the proposed CS-SMATS model are discussed below.

- Applicable to summarize both single document & multi-document.
- Readability & verbal quality of the produced summary.
- Complete coverage of content of the document.
- Non-redundancy of the produced summary.

The effectiveness of the model proposed is proved after comparison with few other nature-inspired optimization algorithms like ACO, BA, FA & FPA. The outcome of the summaries produced by these algorithms is represented in the form of ROUGE score, Range, Co-Variance statistics & Readability metrics over DUC datasets. Out of the above five optimization algorithms, CS-SMATS algorithm-based model performs exceptionally well than the others.

References

[1] S.P. Pati, R. Rautray, An Empirical Analysis of Similarity Based Single Document Summarization, 2021 5th International Conference on Computing Methodologies and Communication, ICCMC). 2021, pp. 860–864, https://doi.org/10.1109/ICCMC51019.2021.9418297.
[2] R. Rautray, R.C. Balabantaray, An evolutionary framework for multi document summarization using Cuckoo search approach, MDSCSA, Applied computing and informatics 14 (2) (2018) 134–144, https://doi.org/10.1016/j.aci.2017.05.003.
[3] K.K. Mamidala, S.K. Sanampudim, A novel framework for multi-document temporal summarization (MDTS), Emerging Science Journal 5 (2) (2021) 184–190, https://doi.org/10.28991/esj-2021-01268.

[4] R. Rautray, R.C. Balabantaray, R. Dash, R. Dash, CSMDSE-cuckoo search based multi document summary extractor: cuckoo search based summary extractor, Int. J. Cognit. Inf. Nat. Intell. 13 (4) (2019) 56–70, https://doi.org/10.4018/IJCINI.2019100103.
[5] Z.H. Ali, A.A. Noor, M.A. Jassim, VIKOR Algorithm Based on Cuckoo Search for Multi-Document Text Summarization, International Conference on Applied Computing to Support Industry: Innovation and Technology, Springer. 2019, pp. 57–67, https://doi.org/10.1007/978-3-030-38752-5_5.
[6] R. Rautray, R.C. Balabantaray, Bio-inspired approaches for extractive document summarization: a comparative study, Karbala International Journal of Modern Science 3 (3) (2017) 119–130, https://doi.org/10.1016/j.kijoms.2017.06.001.
[7] R. Rautray, R.C. Balabantaray, CSTS: Cuckoo Search-Based Model for Text Summarization, Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, Singapore. 2017, pp. 141–150, https://doi.org/10.1007/978-981-10-3174-8_13.
[8] M. Tomer, M. Kumar, Multi-document extractive text summarization based on firefly algorithm, Journal of King Saud University-Computer and Information Sciences 34 (8) (2021) 6057–6065, https://doi.org/10.1016/j.jksuci.2021.04.004.
[9] R.Z. Al-Abdallah, A.T. Al-Taani, Arabic Text Summarization Using Firefly Algorithm, 2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE. 2019, pp. 61–65, https://doi.org/10.1109/AICAI.2019.8701245.
[10] Z.H. Ali, Multi-document summarization using fuzzy logic and firefly algorithm, Journal of College of Education 1 (3) (2017) 139–152.
[11] X. Yang, X. He, Firefly algorithm: recent advances and applications, International journal of swarm intelligence 1 (1) (2013) 36–50, https://doi.org/10.1504/IJSI.2013.055801.
[12] N. Krishnan, G. Deepak, KnowSum: Knowledge Inclusive Approach for Text Summarization Using Semantic Allignment, 2021 7th International Conference on Web Research, ICWR) IEEE. 2021, pp. 227–231, https://doi.org/10.1109/ICWR51868.2021. 9443149.
[13] S. Dhivya, R. Arul, Improved Flower Pollination Algorithm-Based Optimal Placement and Sizing of DG for Practical Indian 52 Bus System, 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), IEEE. 2021, pp. 1–5, https://doi.org/10.1109/IEMTRONICS52119.2021.9422562.
[14] H.O. Sidi, R. Ellaia, E. Pagnacco, A.T. Zeine, Multiobjective Design Optimization Using Flower Pollination Algorithm, 2021 7th International Conference on Optimization and Applications (ICOA), IEEE. 2021, pp. 1–4, https://doi.org/10.1109/ICOA51614.2021.9442651.
[15] R. Sudha, D. Murali, Investigation of maximum observability of standard test bus systems with phasor measurement units using flower pollination algorithm, Int. J. Commun. Syst. 34 (4) (2021) e4711, https://doi.org/10.1002/dac.4711.
[16] H.Q. Cao, H.X. Nguyen, T.N. Tran, H.N. Tran, J.W. Jeon, A robot calibration method using a neural network based on a butterfly and flower pollination algorithm, IEEE Trans. Ind. Electron. 69 (4) (2021) 3865–3875, https://doi.org/10.1109/TIE.2021.3073312.
[17] A. Pattanaik, S. Sagnika, M. Das, B.S.P. Mishra, Extractive Summary: An Optimization Approach Using Bat Algorithm, Ambient Communications and Computer Systems, Springer, Singapore. 2019, pp. 175–186, https://doi.org/10.1007/978-981-13-5934-7_16.
[18] C.X. Yang, J. Zhang, M.S. Tong, A hybrid inversion method based on the bat algorithm for microwave imaging of two-dimensional dielectric scatterers, Progress In Electromagnetics Research M 102 (2021) 91–104, https://doi.org/10.2528/PIERM21012803.
[19] B. Alsalibi, L. Abualigah, A.T. Khader, A novel bat algorithm with dynamic membrane structure for optimization problems, Appl. Intell. 51 (4) (2021) 1992–2017, https://doi.org/10.1007/s10489-020-01898-8.
[20] M. Sipos, Z. Klaic, E.K. Nyarko, K. Fekete, Determining the optimal location and number of voltage dip monitoring devices using the binary bat algorithm, Energies 14 (1) (2021) 255, https://doi.org/10.3390/en14010255.
[21] M.O. Okelola, S.A. Salimon, O.A. Adegbola, E.I. Ogunwole, S.O. Ayanlade, B.A. Aderemi, Optimal siting and sizing of D-STATCOM in distribution system using new voltage stability Index and bat algorithm, 2021 International Congress of Advanced Technology and Engineering (ICOTEN), IEEE. 2021, pp. 1–5, https://doi.org/10.1109/ICOTEN52080.2021.9493461.
[22] A. Al-Saleh, M.E.B. Menai, Ant colony system for multi-document summarization, Proceedings of the 27th International Conference on Computational Linguistics (2018) 734–744.
[23] I.W.A. Setyadi, D.C. Khrisne, I.M.A. Suyadnya, Automatic text summarization menggunakan metode Graph dan ant colony optimization 17, 2018, pp. 124–130, https://doi.org/10.24843/MITE.2018.v17i01.P17.
[24] Z. Gao, J. Zhu, H. Huang, Y. Yang, X. Tan, Ant colony optimization for UAV-based intelligent pesticide irrigation system, 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD). 2021, pp. 720–726, https://doi.org/10.1109/CSCWD49262.2021.9437825.
[25] M. Gao, B. Hu, B. Liu, N. Qiu, H. Zhang, Constrained Path-Planning Control of Unmanned Surface Vessels via Ant-Colony Optimization, 2021 40th Chinese Control Conference (CCC), IEEE. 2021, pp. 4079–4084, https://doi.org/10.23919/CCC52363.2021. 9550270.
[26] M. Shokouhifar, FH-ACO, Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing, Appl. Soft Comput. 107 (2021) 107401, https://doi.org/10.1016/j.asoc.2021. 107401.
[27] P.J. Singh, P. Gupta, D. Singh, Implementing ant colony optimization on evolution obstacle for scheduling a software project, 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), IEEE. 2021, pp. 97–101, https://doi.org/10.1109/ICIEM51511.2021.9445366.
[28] A. Kishor, C. Chakarbarty, Task offloading in fog computing for using smart ant colony optimization, Wireless Pers. Commun. 127 (2021) 1683–1704, https://doi.org/10.1007/s11277-021-08714-7.
[29] M. Peyrard, J. Eckle-Kohler, A general optimization framework for multi-document summarization using genetic algorithms and swarm intelligence, Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers. 2016, pp. 247–257.
[30] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, DESAMC+DocSum: differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization, Knowl. Base Syst. 36 (2012) 21–38, https://doi.org/10.1016/j.knosys.2012.05.017.
[31] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, CDDS: constraint-driven document summarization models, Expert Syst. Appl. 40 (2) (2013) 458–465, https://doi.org/10.1016/j.eswa.2012.07.049.