Received: December 28, 2020. Revised: March 10, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Hybrid Algorithm Based on Chicken Swarm Optimization and Genetic Algorithm for Text Summarization

Mostafa Gamal[1]* **Ahmed El-Sawy[1]** **Ahmed H. AbuEl-Atta[1]**

[1]*Computer Science, Faculty of Computers and Artificial Intelligence, Benha University, Egypt*
* Corresponding author's Email: mostafa_gamal_1111@yahoo.com

Abstract: The daily massive flow of information requires automated summarization methods to extract the most important information. Manual summarization of large text documents is very complicated and time-intensive for human beings. Numerous methods rate all of them decadently based on the sentence scoring that labels ratings for input sentences. The higher-rated sentences are used as a part of the summary. In an extractive-based automated text summary, locating the relevant sentences is an essential problem for the researchers. Therefore, to deal with such problems, evolutionary algorithms are applied as a solution. This paper presents a hybrid approach (CSOGA) based on the effectiveness and convergence to the solution of a (CSO) chicken swarm optimization and a (GA) genetic algorithm for text summarization to ensure the optimal solution. The evaluations of the proposed algorithms are done on the standard dataset from CNN / Daily Mail and are measured by the Recall-Oriented Understudy for Gisting Evaluation (ROUGE). The performance of the proposed method is then compared with other methods. The results show that the new approach hybrid (CSOGA) has the best performance on text summarization quality. The proposed method was capable of generating a better accuracy than other algorithms on the ROUGE-1, ROUGE-2 and ROUGE-L. The highest increase in the accuracy of the proposed method was in ROUGE-1 with a rise of 4.4%, ROUGE-2 with a rise of 12.01%, and ROUGE-L with a rise of 9.8% comparing with the highest accuracy of the other extractive models.

Keywords: Automatic text summarization, Chicken swarm optimization, Genetic algorithm, Artificial bee colony.

1. Introduction

The World Wide Web provides an enormous amount of content in web page forms, news articles, documents, email, and access to databases around the world. That makes nominating the most relevant information through manual means an extremely difficult process. Text summarization is one of the methods that can overcome the huge amount of information. The summary approach removes the time and hard work needed to find the most important and relevant sentences in the documents. A summary can generally be defined as a text generated from one or more texts which, while being short enough, conveys the most valuable details in the original.

As a result, automated text synthesis has become a research area in the wider field of language analysis to help locate essential documents [1]. The text summary aims to extract the most relevant elements of the given document. The subsequent definitions from underlying assumptions regarding text summarization. A summary is a text created from one or more texts are containing a significant part of the knowledge found in the original texts, and not more than half of the original texts [2]. Text summarization can be defined as the following: given a document X, which is considered as a group of sentences such that, X = {Se1, Se2,..., Sen}, where Sei indicates ith sentence in X and n is several sentences within the document. The summary of document X is the subset Y = {Se (1), Se(2),..., Se(m)})} ⊂ X where m < n, which contains the different topics of document X while reducing the redundancy within the summary Y.

Text summarization techniques are classified into abstract and extract related techniques according to specific requirements. An abstract can be defined as a description of the ideas taken from the root are then reinterpreted and brought out different. The extract is a list of units of text taken from the source and displayed inline for line form.

The remainder of the article is structured as follows. Section 2 introduces the literature review, section 3 presents preprocessing and how to select features, section 4 introduces the proposed algorithms, and the results of evaluation using CNN / Daily Mail dataset are represented in section 5. Section 6 presents the discussion, section 7 presents the conclusion, and section 8 presents' references.

2. Literature review

There are several approaches for a single document text summarization in the literature. These techniques are categorized according to a variety of methods including semantic-based, statistical-based, machine-based, cluster-based, graph-based, and optimization-based methods [3-4].

Semantic analysis has a great deal to do with the meaning of words as well as the relations between terms, and phrases to construct the intended meanings of the text. Several semantic analytical methods may use to summarize documents, including lexical chains, and natural language processing strategies, such as latent analysis [5,6] have utilized users who used question keywords or topics to produce a summary of the original text. Sentences with the highest score are chosen as summary. These methods trying to improve the text summarization quality by generating more informative and less redundant summaries, which is a difficult task because it has difficulties in utilizing high-quality semantic research methods.

Statistical approaches are commonly used in summarizing texts. The concept of relevance score which depends on the extraction of a collection of attributes, is the deciding element that represents the importance of a statement, regardless of its context. In [7] the compilation of sentences relies on the retrieval of main phrases. The derived key-phrases are based on characteristics such as Term Frequency (TF), Inverse Document Frequency (IDF), Font Forms, and their appearance in the title from the paper. The key phrases selected will then be checked for their ability to reflect the meaning of the sentence. In [8-9] authors used other features to score the sentence including indicator phrases, uppercase words, sentence length, the similarity with the title, and sentence position in the document. For sentence ranking, a weighted linear combination of statistical characteristics is used in [10]. They also obtained the optimum weights using a genetic algorithm. These features are simple to incorporate and can be used to improve main sentence collection or to reduce duplication. The weak point of these features is their independence on the meaning of the text because they often rely just on statistical measurements.

Machine learning and deep learning methods have been applied in the recent past to the task of summarizing text and have achieved a high rate of success. Models ranging from basic multi-layer networks [11] to complex architectures of the neural network [12] are suggested for text summary. Machine learning approaches are very efficient and successful in a single and multi-document description. However, a series of training documents (marked data) is required to train the classifier. Furthermore, Machine learning approaches outputs are influenced by the selected classifiers, attributes, and representation attributes which play an important role in their success.

The Clustering process aims at grouping objects into classes drawing on the similarities. When summarizing documents, the objects are the sentences, groups are clusters to which sentences belong. In this method, the description shall be drawn up by choosing a sentence or more from each cluster depending on the vicinity of the cluster centroid [13]. Clustering strategies are used to minimize data duplication by categorizing similar data, but the description produced may not be significant enough. This happens because the chosen sentences are primarily graded according to the proximity of the centroid cluster. These sentences are computed by distance calculation without paying any attention to the nature of the sentence or centroid document.

Graph method in this model the nodes of the graph represent the sentences while the links/edges between the related nodes represent the similarity of the sentences. Therefore, a sentence is considered relevant if it is closely related to several other sentences [14]. LexRank [15] and TextRank [16] are two well-known graph-based rating schemes that are used in the graph-based approach. The use of graph-based approaches has a positive influence on multi-document analysis communities as it has the potential to identify different themes from unconnected sub-graphs. However, the creation of sub-graphs, based on the calculation of statistical similarity, without paying any attention to the text meaning, which risks the production of a less-informative summary [17].

It was, therefore, important to find a way to get the best relative to previous methods, from the present-day strategies in this field that depend on the hypothesis of swarm insights, a few calculations have been connected in this region to get good results. For example in [18] the cuckoo search algorithm with sentiment score is applied for text summarization and by the experiments it had good results. In [19] the author's used particle swarm optimization algorithm for text summarization for Hindi language and the result compared with other algorithms and had good result rather than a genetic algorithm and other algorithms. In [20] there are hybrid approach contains genetic algorithms and domain knowledge for Arabic text summarization from experiment get good results. So that in this paper bio-inspired swarm algorithms are applied for text summarization for English single documents. The results proved the quality of these algorithms in obtaining the solution because of its advantages. We designed an extractive model for text summarization based on chicken swarm optimization and genetic algorithm to achieve the quality of meaning and optimization and to solve the drawback of other algorithms such as quality and time consumption.

We created an extractive model for summarization based on the quality of meaning and optimization. To ensure the quality of the meaning of the summary, we find the extent to which each sentence is related to the other in the document instead of extracting the sentence with the highest value, and then it is a turn to methods provide an optimization model for choosing the relevant sentences from each category to construct an optimal description.

In the experiments, we use CNN / Daily Mail dataset to evaluate the proposed model. The results show that our model outperforms several strong comparisons models in quality of summary and speed to reach the solution

The main contributions of this paper are as follows:

- Simplicity, flexibility, robustness, and the ability to explore local solutions are among where to pick TSABC model proposed in the paper gives priority to get high quality of summary, but it takes a lot of time to reach a solution.
- The TSCSO model proposed in the paper gives priority to reducing the time to reach the summary, particularly in large documents due to the properties of the chicken swarm optimization algorithm quick interpretation, simple process, and fast search, but one of the disadvantages may stuck in a local optimum by processing a single document.
- The CSOGA show proposed in the paper the results proved that the issue of TSCSO may be stuck in a local optimum. The genetic algorithm solve this issue through a crossover and mutation process and one of the points of interest the algorithm is high efficiency to get the highest quality of the summary in the quickest time.

3. Pre-processing and selected features selection

The pre-processing of the text is a required step because the consistency of the summaries generated is dependent on pre-processing and representation of specific text. There are four key tasks to be carried out in this stage:

Sentence Segmentation is a natural-language processing task. It is used to locate each sentence in a separate line within the document. It also concerns the detection of the sentence boundary, which includes many notation marks that share the characteristic endpoints of sentences such as.",?" "and!."

Tokenization separates the document input into single words. There are many notation marks used to discover the boundary of tokens such as the tab, white space, period, colon, semicolon, and comma, etc.

Stop words removal is used to remove the words that do not affect text relevance.

Words Stemming is the process by which each word returns to its root form.

Many features are extracting from the document and the proposed algorithms applied these features for text summarization each feature plays an important role in the document summarization process.

Title Feature (TF): A sentence containing any of the "Title" words is considered an important and topic-related sentence. TF can be calculated using Eq (1).

$$TF(Si) = \frac{CountWord(si) \cap CountWord(Title)}{CountLength(Title)} \quad (1)$$

Where CountWord() is a function used to count input sentence terms, of the input parameter such as the sentence in the document that is intersected with the Title words; CountLength() is a function that computes the length of the title based on the number of words enclosed.

Sentence Length (SL): To prevent selecting sentences that are either too short or too long, a normalized division may solve the issue. Eq (2) shows such a normalization, where Si refers to the ith sentence in the document consists of n words, Sj referring to the longest sentence in the document consists of m words, and CountLength() is a function that computes the length of each input sentence based on the number of words found [21].

$$SL(Si) = \frac{CountLength(S_{(i,w\in\{1,2,\dots,n\})})}{CountLength(S_{(j,w\in\{1,2,\dots,m\})})} \quad (2)$$

Sentence Position (SP): The first sentence in the paragraph is a valuable sentence and a good candidate for addition to the summary. Eq (3) is used to calculate the SP feature, Where Si refers to the ith Sentence in the document wanted to extract its position score, and CountTotal( ) is a function that retrieves the total number of the sentences in the input parameter document d and CurrentPosition( ) is a function that retrieves the current order of sentence Si in document d [21].

$$SL(Si) = \frac{CountLength(S_{(i,w\in\{1,2,\dots,n\})})}{CountLength(S_{(j,w\in\{1,2,\dots,m\})})} \quad (3)$$

Numerical Data (ND): A sentence that contains numerical data also contains essential details such as event date, money expenditure, percentage of damage, and so on. Eq (4) shows how to measure this function.

$$ND(Si) = \frac{CountND(Si)}{CountLength(Words(Si))} \quad (4)$$

Where CountND() is a method that measures the numbers of Numerical Details (ND) appearing in the document's ith sentence S in the document, and CountLength() is a function used to compute the sentence length of Si [23].

Thematic Words (TW): are a list of top n selected terms with the highest frequencies. To calculate the thematic words first, the frequencies of all terms in the document are computed. Then, a threshold is specified to assign terms that should be selected as thematic words. In this case, the top ten frequent-terms max(TW) would be assigned as a threshold. To compute the ratio of TW found in the ith sentence S in the document Eq. (5) is used where CountThematic() is a function used to compute the number of the thematic words found in sentence Si [22,24].

$$TW(Si) = \frac{CountThematic(Si)}{max(TW)} \quad (5)$$

Final scoring for the sentence: The ranking of the sentence represents the term strength. Using Eq(6), the final score for each sentence is calculated using a linear combination of all previously derived attributes.

$$Score(Si) = TF(Si) + SL(Si) + SP(Si) + ND(Si) + TW(Si) \quad (6)$$

4. The proposed algorithms

The proposed algorithms use the optimization algorithms principle to determine the best combination of sentences for a description based on the coherence of phrases and the insightful ranking. The goal is to collect extremely informative performance sentences with good coherence,

Where the informative performance can be calculated using the weight of the function and the similarity can be used to calculate the cohesion of the sentence. The Popular parameters are listed below between the three proposed optimization algorithms.

4.1 The initial population

Is chosen at random where the population is a set of a positive integer representing the sum of the original text message. The group (row) is shown as ($Se_{1,1}$, $Se_{1,2}$, $S_{e1,3}$...$Se_{1,m}$). Each category represents a review of the nominees. The group-length is the summary-length equivalent. One summary sentence is the group element.

4.2 Evaluate the fitness

The object of the fitness function is to optimize the mean similarity between sentences with a higher insightful score, so it can be written using Eq (7). The matrix displays the connection between the sentences of the text.

$$Sim(sj, sk) = \frac{\sum_{i=1}^{t}(wi,j)(wi,k)}{\sqrt{\sum_{i=1}^{t} w(i,j)^2}\sqrt{\sum_{i=1}^{t} w(i,k)^2}} \quad (7)$$

Where the cosine relation Sim(sj,sk) is both sentence j and sentence k, (wi,j)is the weight of the term I in the jth paragraph, and t is the number of terms in the sentence [25].

Through Eq (8) can calculate the final score of sentences.

$$Fitness\ function = \sum_{i=1}^{n}\sum_{j=i+1}^{n} Score(Si)Sim(sj, sk) \quad (8)$$

4.3 criteria to Quite

The operations are replicated, as seen in the overview extraction algorithms before the full number of iterations is reached. This chooses the agent (rank) with the maximum value as the number extracted from the final community.

4.4 Text Summarization based on Artificial Bee Colony Algorithm (TSABC)

Simplicity, flexibility, robustness, and the ability to explore local solutions are among were to pick this algorithm. The colony is composed of three classes of bees in the Artificial Bee Algorithm model: working bees, spectators, and scouts. Scouts conduct spontaneous searches, working on bees to gather food previously discovered, and audiences observe the movements of the bees and pick food sources depending on the movement. Bee-to-bee contact is dance-based. The number of food sources within the population reflects several solutions. The origin of the food source is a potential solution to the problem of optimization, and the quantity of nectar in the food source is correlated with the solution's consistency (fitness) [27].

Initialization Phase: The definition below may be used for initialization

$$X_{mi} = \mathrm{l}_i + rand(0,1)(u_i - \mathrm{l}_i) \tag{9}$$

Where l_i and u_i , are the lower and upper limit of $the\ X_{mi}$ function respectively.

Employed bees phase: look for new sources of food that have more nectar in their memory within the food source neighborhood. They find a source of food for their neighbors. For this step, the description below may be used.

$$\mathrm{v}_{mi} = x_{mi} + \phi_{mi}(x_{mi} - x_{ki}) \tag{10}$$

Where x_{ki} is a source of food chosen at random. ϕmi is the random number within the [−a, a] range. If the latest food supply has been developed the health is determined.

Onlooker Bees Phase: The probability value pm selected by an onlooker bee can be determined using the description below

$$P_m = \frac{fit_m(\mathrm{x}_m)}{\sum_{m=1}^{sn} fit_m(\mathrm{x}_m)} \tag{11}$$

Using Eq (9) a neighborhood source is determined and its fitness value is evaluated after a probabilistic collection of the food supply for an onlooker hive. As with the bees working in the process, there is a gripping competition between them. Therefore more spectators are drawn to richer sources and there is positive feedback activity.

```
Algorithm 1: TSABC
Data:    Document : D,
         Size of the population: M,
         Number of iteration: N,
         Number of food sources: M/2,
         Limitation Number: limit
Result:  Summary

Pre-processing on D;
Eq (1-6) are applied to get list of weighted sentences;
Generate sentences indexes as feasible solutions (Initialization)
    randomly as P = {$x_i$} ,i= 1,......,n ;
Keep them in the population Pop;
Calculate fitness F for each sentence in pop using Eq (8);
for t = 1 to N do
    for i = 1 to M/2 do                                  (Employed bee stage)
        Generate indexes of new sentences as new solution vi use Eq (10);
        if fit($v_i$(t+1)) ≥ fit($x_i$(t)) then
            $x_i$(t+1) ← $v_i$(t+1)
        else
            $x_i$(t+1) ← $x_i$(t)
    Using equation(11) calculate the probability values pi for Xi solutions;
    for i = 1 to M/2 do                                  (Onlooker bee stage)
        if r < $p_i$ then
            Generate new sentences indexes as new solution $v_i$ using Eq (10);
            if fit($v_i$(t+1)) ≥ fit($x_i$(t)) then
                $x_i$(t+1) ← $v_i$(t+1)
            else
                $x_i$(t+1) ← $x_i$(t)
    if limit is reached then                             (Scout bee stage)
        Reinitialize the population;
        Evaluate population P = {$x_i$} ,i= 1,......,M using Eq (8);
    Remember the so far strongest solution;
endfor
return best solution
```

When applying Algorithm 1 on any document the same preprocessing and feature selection will apply when running the algorithm three-stage working like employed in this stage the algorithm begins to generate new solution meaning find sentences after that calculate the fitness of it and comparison it with the sentence in initial population and choose the high score, onlooker in this stage by using the Eq (11) find the probability of each solution and calculate the fitness after that comparing result and save the best solution, and scout until to obtain the best summary.

Text Summarization based on chicken Swarm Optimization Algorithm (TSCSO)
design of the speed reducer can be effectively solved by CSO, which induces a promising prospect of further study from CSO. One of the reasons why CSO performs so promisingly is that CSO inherits major benefits from other algorithms [29]. Various chickens' movements may be beneficial to discovering the algorithm equilibrium to reach a good balance between randomness and resolution. The chicken swarm is split into groups and each group has the main rooster, a few hens, and chicks. Within society, the mother-to-child bond and friendship will remain unchanged. Just refresh a few (G) days. Chicken pursues their group mates' rooster to collect food to deter them from consuming their meals.
Chickens Roosters' action has a preference for consuming food with better fitness values than those with poor fitness values. This can be spelled out below.

$$\mathrm{x}_{i,j}^{t+1} = \mathrm{x}_{i,j}^{t}\left(1 + rand(0, \sigma^2)\right) \tag{12}$$

$$\sigma^2 = \begin{cases} 1 \quad , & if\ f_i \leq f_k \ \ k \in [1, N], k \neq i \\ exp\left(\dfrac{f_k - f_i}{|f_i| - \varepsilon}\right), otherwise & \end{cases} \tag{13}$$

Where Rand (0, σ^2) Is an average Gaussian distribution of 0 and σ^2 . ε, this is the smallest constant computer which is used to avoid zero division error, k the rooster index, the fitness assessment of the corresponding x is randomly selected from the rooster group f. The more dominant hens, when competing for food, would have an advantage over the more submissive ones.
These phenomena can be mathematically formulated as follows.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S1(Rand\left(x_{r1,j}^{t} - x_{i,j}^{t}\right)) + S2(Rand\left(x_{r2,j}^{t} - x_{i,j}^{t}\right)) \tag{14}$$

$$S1 = \exp\left(\frac{f_i - f_{r1}}{|f_i| - \varepsilon}\right) \tag{15}$$

$$S2 = \exp(f_{r2} - f_i) \tag{16}$$

Where Rand is a uniform random choice over [0, 1]. r1 [1, .., N] is the rooster index, which would be the team-mate of the ith hen, whereas r2 [1, .., N] is the chicken index (rooster or hen), randomly selected from the swarm. r1, r2.

Algorithm 2: TSCSO

Data:	Document : D, Size of the population : P, MaxGeneration: MG Number of groups: G
Result:	Summary

```
Pre-processing on D;
Using eq (1-6) are applied to get a list of weighted sentences;
Generate sentences indexes as feasible
 solutions randomly as P = {x_i} ,i= 1,......,n ;
Keep them in the population Pop;
Calculate fitness F for sentences in pop by eq (8);
While (t < MG) do
    if (t % G == 0)
        Rank each sentence in pop according
         a score of the sentence;
        Establishes hierarchical order A, B, C;
        Divide the pop into different groups, the relationship is determined;
    end
    for i = 1 to P do
        if i == A then
            Update sentence index and its score
             using eq (12);
        if i == B then
            Update sentence index and its score
             using eq (14);
        if i == c then
            Update sentence index and its score
             using eq (17);
        endif
    Evaluate the current best score
    if current score ≥ global score then
        global score ← current score
end
```

The chicks move around to feed their mother forage. This is written out below.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL\left(x_{m,j}^{t} - x_{i,j}^{t}\right) \tag{17}$$

By applying Algorithm 2 on any document. Firstly some pre-processing applied to the document like sentence segmentation, tokenization, stop-words removal, and word stemming. Second step using Eq (1), Eq (2), Eq (3), Eq (4), Eq (5), and Eq (7) to prepare these features for use later. The third step initializes random solutions from sentences and runs the algorithm, establishes a hierarchical order that is the behavior of the algorithm, and will update this hierarchy after a specific iteration. The algorithm runs until to obtain the best summary

4.5 Text Summarization based on hybrid Algorithm (CSOGA)

The CSO has advantages such as quick interpretation, simple process, and fast search. However, to solve a large complex problem, CSO may be stuck in a local optimum. This deficiency must be resolved to maximize the practicality of the CSO. GA has advantages such as low control parameters and high efficiency, but it also has some flaws, such as slow convergence speed and low text summary accuracy. In CSO, high randomness of roosters, hens, and chicks movements lets the quest process hop rapidly from one area to another. The global search functionality of the algorithm is therefore very high. However, considering the high randomness of the gestures, the algorithm initiates a blind search procedure, the search efficiency is dramatically reduced close to the optimal solution. To boost the efficiency of the CSO, GA is implemented in the CSO upgrade process. A CSOGA hybrid algorithm is therefore being developed. CSOGA first uses the movement of roosters, hens, and chicks in the search space to search, and then uses the CSO update mode location to accelerate the convergence of swarms to the maximal solution. At the same time, the random exclusion process of GA will successfully avoid local optimization through crossover and mutation process, thereby improving the efficiency of the search for an optimal solution. Fig. 1 explained the main step of the algorithm.

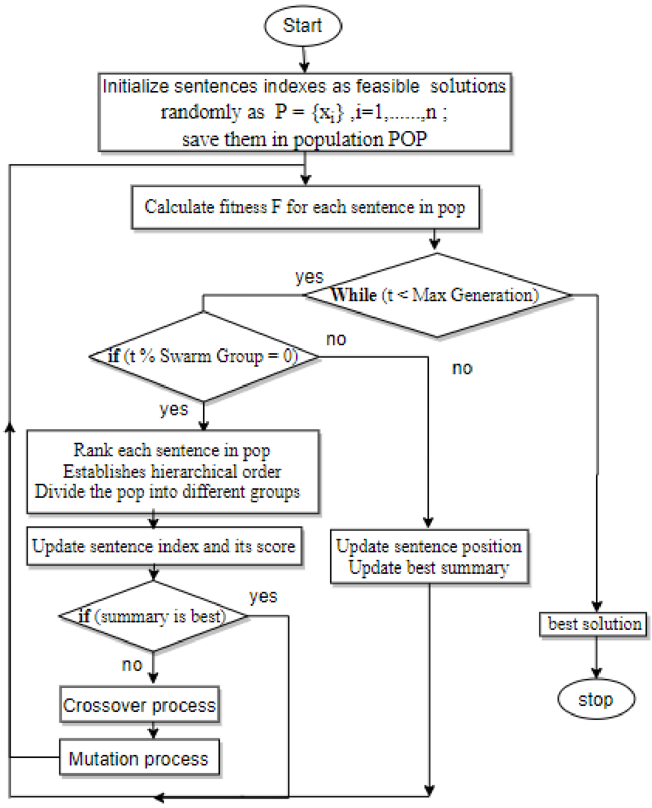
Start
Initialize sentences indexes as feasible solutions randomly as P = {x_i} ,i=1,.......,n ; save them in population POP
Calculate fitness F for each sentence in pop
yes
While (t < Max Generation)
no
if (t % Swarm Group = 0)
no
yes
Rank each sentence in pop
Establishes hierarchical order
Divide the pop into different groups
Update sentence index and its score
Update sentence position
Update best summary
yes
if (summary is best)
no
best solution
stop
Crossover process
Mutation process

Figure. 1 Flowchart of CSOGA

There are many phases in the hybrid approach CSOGA:

Phase 1. The parameter is set, the population is initialized. The parameters Document: D, Size of the population: P, t: iteration, MaxGeneration: MG, Swarm group: G, Rate of crossover: C, and Rate of mutation: M are set .population is initialized randomly.

Phase 2. The initial population of sentences indexes fitness value is measured using the objective function and assign the weight for each sentence.

Phase 3. The chicken swarm algorithm is initiated. The position of the individuals (roosters, hens, and chicks) represent sentence indexes are updated by using Eq (12) for rooster, Eq(14) for hen, and Eq(17) for chicks, and a new individual of the sentence is produced. If the iteration mod swarm group equal 0 will rank each sentence after that the fitness values of the new solution (summary) and old individuals are compared; if the summary is better continue else go to phase 4.

In this phase as the proposed algorithm of TSCSO, although the algorithm gives priority to reducing the time to reach the summary, one of the disadvantages may be stuck in a local optimum by processing a single document. As shown in Fig.7 and Fig.8 this represents a problem that must be solved to ensure that the optimal solution is obtained. We block the algorithm from stuck in a local optimum as shown in phase 4.

Phase 4. Genetic algorithm [26] is initiated and uses algorithm [26] crossover through select two summary S1, S2 randomly from Pop. The second step Generate S3 and S4 by one-point crossover to S1 and S2. Then Keep S3 and S4 in the new population pop1 .after that apply the process of mutation through select a summary Sj from Pop1. Mutate each bit of Sj under the rate M and generate a new summary Sj ′ . And so we made sure the algorithm may be stuck in local optimum the process of crossover and mutation will block the algorithm from the stuck. Also, the algorithm is efficient in obtaining quality results for the summary

Phase 5. Global and individual optimal values are modified. The best locations of all the individuals and entire populations are modified.

Phase 6. If iteration equals max generation then the algorithm is satisfactory, then the optimal summary is generated, and the algorithm is terminated; otherwise, Step 3 is performed.

5. Experimental results dataset

Received: December 28, 2020. Revised: March 10, 2021.

CNN/Daily Mail dataset is the most widely used developed by Nallapati [30]. The dataset includes online news articles (average 781 tokens) coupled with multi-sentence summaries (an average of 3.75isentences or 56 tokens). The controlled version containsi287,226 training pairs, testing pairs 13,368, and check pairs 11,490. Models are verified with full-lengthiF1-scores of ROUGE-1, ROUGE-2, and ROUGE-L. In the field of text summarization, there are many algorithms applied, such as algorithms in Table 2, and used this dataset in this paper used this dataset and applied it to the proposed algorithms.

Metric evaluation:

We're using ROUGE as our assessment metric. ROUGE checks the accuracy of the summary by computing the same lexical units, such as unigram, bigram, trigram, and the longest general sequence (LCS). The summary evaluation has become common. Following the previous work, ROUGE-1 (unigram), ROUGE2 (bigram), and ROUGE-L (LCS) are used as evaluation criteria in the analysis findings referred.

Experiments are done on the metaheuristic algorithms (TSCSO), (TSABC), and hybrid algorithm (CSOGA) using different sizes (the number of sentences) of the documents. Each algorithm run and each run contains many iterations and to find the best population size in each algorithm.

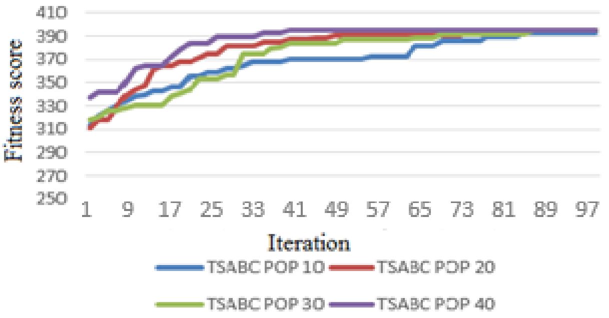

Figure. 2 Scoring of the different population size of TSAB

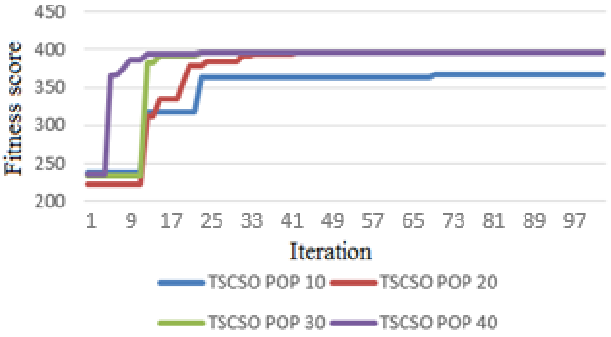

Figure. 3 Scoring of the different population size of TSCSO

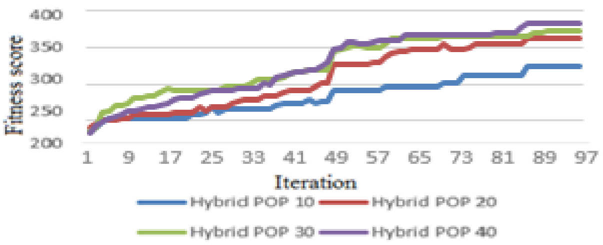

Figure. 4 Scoring of the different population size of CSOGA

Fig. 2, 3, and 4 showed that for all proposed algorithms the best population size. We will select according to the best fitness score assigned. In the first algorithm, it is clear that the TSABC algorithm with a population size equals to 40 gives the best solution and reaches the solution faster than other population sizes. In the second experiment of different population sizes of the TSCSO algorithm as shown in Fig. 3, the results showed that with a population size of 40 the algorithm needs only 20 iterations to reaches the best score. Third approach as shown in Fig. 4. Each population size will increase the fitness score in each iteration until reaches the maximum solution but from notice when increase the population size performs well in the first iterations.

Table 1 contains the result of the experiments, there are three types of documents (small, medium, large) the best, worst, average, and standard deviation value of each algorithm. In Fig. 8 represent the average score of fitness the result in small document TSABC is the best and other sizes TSCSO assign the good value in medium and large size comparing to TSABC and CSOAG is the best of them.

Table 1 Scoring of fitness of each algorithm in various size of document

Doc.	Method	Best	Worst	Avg.	Std. dev.
DOC1	TSABC TSCSO CSOGA	273.5 270.34 274.33	190.4 192.0 194.88	262.7 265.3 270.8	**6.56** **31.34** 21.35
DOC2	TSABC TSCSO CSOGA	380.8 383.8 430.1	276.2 180.5 253.6	371.1 365.0 380.3	33.98 7.47 24.42
DOC3	TSABC TSCSO CSOGA	530.9 569.8 575.31	395.6 531.8 356.21	548.1 544.6 556.8	42.67 13.69 24.84

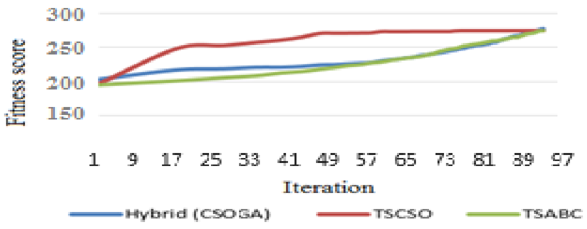

Figure. 5 Scoring of algorithms on small size document

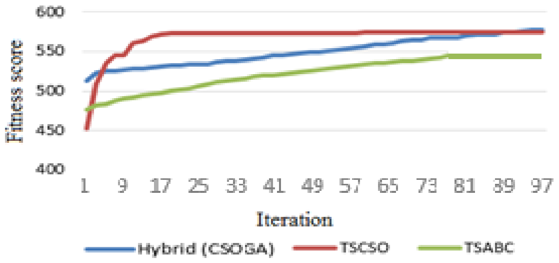

Figure. 6 Scoring of algorithms on medium size document

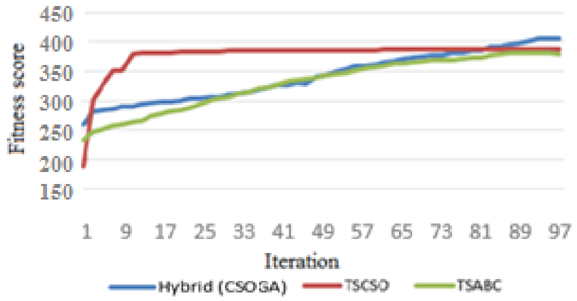

Figure. 7 Scoring of algorithms on large size document

The proposed algorithms are applied on small size document as shown in Fig.5 and the results show that the convergence to reach the high score for the TSCSO algorithm but TSABC and CSOGA algorithms need more iteration to settle on the optimum solution. As shown in Fig.9 the time consumption of each algorithm in the experiment on small documents The TSCSO need less time to generate the summary but CSOGA need more time but the quality of summary higher than other algorithms as shown in Table 2. In the experiment of the medium document as shown in Fig.6, we noticed that TSCSO needs less iteration to reach the solution. On the other side the other two algorithms need more iterations but the result better than TSCSO. The algorithms applied on large document the results showed that TSCSO better than TSABC and the CSOGA is the best. Also as shown in Fig.9 the time consumption of CSOGA need more time in all document and TSCSO needs less time.

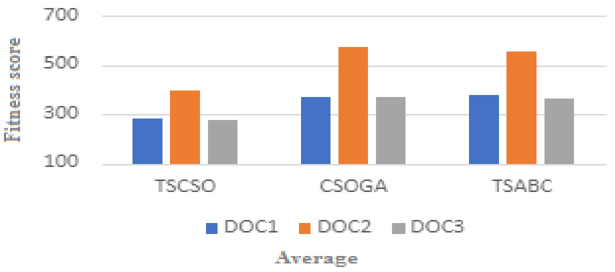

Figure. 8 The average score of each algorithm applied to different documents

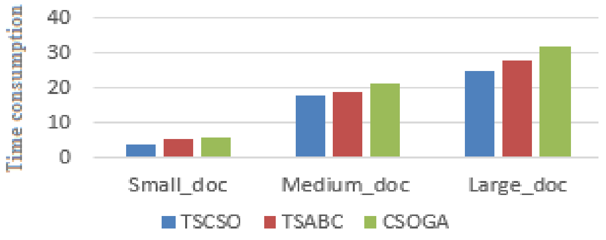

Figure. 9 The time complexity of proposed algorithms

Table 2. ROUGE results for comparison CSOGA and systems are taken from the authors' respective papers.

Models	**ROUGE-1**	**ROUGE-2**	**ROUGE-l**
LEAD-3	40.34	17.70	36.57
ORACLE	52.59	31.24	48.87
Abstractive			
PTGEN 2018 [30]	36.44	15.66	33.42
PTGEN+COV 2018 [31]	39.53	17.28	36.38
DRM 2018 [32]	39.87	15.82	36.90
BOTTOMUP 2019 [33]	41.22	18.68	38.34
DCA 2019 [34]	41.69	19.47	37.92
Extractive			
REFRESH 2018 [35]	40.00	18.20	36.60
LATENT 2018 [35]	41.05	18.77	37.54
NEUSUM 2018 [36]	41.59	19.01	37.98
SUMO 2019 [37]	41.00	18.40	37.20
LSTM+PN 2017 [38]	41.85	18.93	38.13
HER w Policy 2017 [39]	42.30	18.90	37.60
Hybrid (CSOGA)	**44.20**	**21.17**	**41.30**
TSCSO	**43.33**	**20.21**	**40.51**
TSABC	41.26	18.77	37.60

We compare our models with the following state-of-the-art. Automatically, we tested the summarization efficiency using ROUGE. We're reporting unigram and bigram overlap (ROUGE-1 and ROUGE-2) as a means of assessing information quality and the longest common subsequence (ROUGE-L) as a method of evaluating fluency.

Table 2 summarises the results of our study on the dataset of CNN / DailyMail. The first block of the table contains the effects of the extractive ORACLE The system is an upper bound. Also, LEAD-

Received: December 28, 2020. Revised: March 10, 2021.

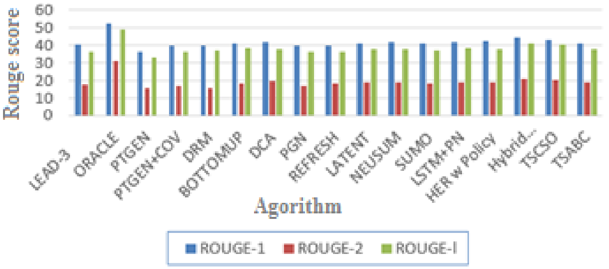

Figure. 10 Comparison between the proposed algorithms and other algorithms as shown in table 2.

3ibaseline selects only the first three sentences in the document.

The second part includes state-of-the-art neural network and statistical methods based extractive models, i.e., PGN, DCA, PTGEN, and BOTTOMUP models

The pointer generator network (PTGEN) enables it to operate copy words from the source text and cover mechanism that keeps track of terms that have been summed up.

Also Model (DRM) for abstract summarization that deals with the problem of coverage of the intra-attention mechanism where the decoder is present over previously generated terms.

Follow method (BOTTOMUP) the content selector first decides which phrases in the source material can be part of the description, and the copy function is then added to the pre-selected text. Phrases during the encoding process.

In the third part, we report the state-of-the-art extractive models i.e., LATENT, NEUSUM, and HER w Policy.

REFRESH is a system based on learning reinforcement

Trained by refining the ROUGE metric worldwide. Recent thesis achieves better efficiency more advanced model systems.

LATENT presents extractive summarization as a latent factor inference problem instead of optimizing the probability of "gold" norm marks, their latent model explicitly maximizes the probability of human summarization sentences selected.

SUMO capitalizes on the notion of structured attention to induce the interpretation of a document in a multi-root dependency tree while predicting the output summary.

NEUSUM scores and collectively selects sentences and represents the state-of-the-art of extractive summarization.

The hybrid algorithm (CSOGA) achieves 21.17 and (TSCSO) algorithm score 20.21 ROUGE-2 F1 score on CNN/Dailymail dataset. Compared to extractive and abstractive models. The result showed that the two algorithms perform better than others. In terms of ROUGE2 F1, (CSOGA) and (TSCSO) outperforms the strong baseline LEAD3 by 3.47 and 2.51 points. Compared to the state-of-the-art extractive models significantly better in terms of ROUGE-1, ROUGE2, and ROUGE-L F1 scores.

The F-measure indicates a summarizer's performance. The better the summarizer's performance is the closer that value to one. While the results were obtained, several points should be noted that the CSOGA algorithm achieved the best f-measure but need much time to reach the best score. According to the results, it is clear that the TSCSO algorithm is faster to reach a solution and has more stability.

6. Discussion

summarization. Two of these algorithms are based on swarm theory and the other is new. The results from the experiments conducted on the database indicated that these algorithms provided very successful outcomes compared to other algorithms. When looking at the results in Table 2 comparing these proposed algorithms with each other, and state-of-art algorithms. The results showed that the proposed method CSOGA was capable of generating a better accuracy than other algorithms on the ROUGE-1, ROUGE-2, and ROUGE-l. The highest accuracy increase of the proposed method was in ROUGE-1 with a rise of 4.4%, ROUGE-2 with a rise of 12.01% followed by a 9.8% for ROUGE-l comparing with the highest accuracy in extractive models. In addition to increasing accuracy by 10.50%, 16.3%, and 12.8% compared with the lowest accuracy in extractive models. The result of TSCSO increased by 2.4%, 6.9%, and 7.7% comparing with the highest accuracy in extractive models. In addition to increasing accuracy by 8.3%, 11%, and 10.6% compared with the lowest accuracy in extractive models. One of the drawbacks of the TSCSO algorithm may be stuck in a local optimum as shown in Fig.6 the algorithm reaches the solution and stay on it from iteration 27. Therefore, there had to be a solution to this problem, and this is through the CSOGA algorithm by genetic algorithm solve this issue through a crossover and mutation process and one of the points of interest the algorithm is high efficiency to get the highest quality of the summary in the quickest time. When comparing CSOGA with TSCSO we find that the accuracy of CSOGA increased by 2%, 4.7%, and 1.9%. The results proved that efficiency of genetic algorithm efficiency to

Received: December 28, 2020. Revised: March 10, 2021.

escape from the local optimum and achieve a good result. The result of TSABC showed some strengths and some weaknesses according to strengths increase the accuracy by 0.6%, 2%, and 1% when comparing with one of the extractive models but most of the achieve score higher than it. One of the weaknesses of time consumption if shown in Fig.9 it consumes not little time for good accuracy, but it is not optimal. To solve this problem, a fast algorithm had to be used like the TSCSO model proposed in the paper gives priority to reducing the time to reach the summary, particularly in large documents due to the properties of the chicken swarm optimization algorithm quick interpretation, simple process, and fast search. When comparing the two algorithms we find that TSCSO is more accurate than TSABC by 5%, 7.6%, and 7.7%. And when solving a problem of TSCSO through CSOGA we find that CSOGA is more accurate than TSABC by 7.1%, 12.7%, and 9.8%. According to time complexity, we noticed in Fig.9 the faster algorism in convergence to reach the solution is TSCSO in all experiments, TSBAC in the second position but CSOGA in the third position because of the process of cross over and mutation of genetic algorithm take much time but high quality. The comparison is done on the CNN / Daily Mail dataset to evaluate the algorithms. The proposed are implemented and compared with state-of-the-art text summarization algorithms. The optimization algorithms showed that the fast to reach the best solution and its ability and performance in text summarization.

7. Conclusion

This paper presents the hybrid optimization algorithm CSOGA for text summarization, which is based on the chicken swarm algorithm, and the genetic algorithm.

The highest accuracy increase of the proposed method was in ROUGE-1 with a rise of 4.4%, ROUGE-2 with a rise of 12.01% followed by a 9.8% for ROUGE-l comparing with the highest accuracy in extractive models.

The chicken swarm algorithm is chosen due to quick interpretation, simple process, and fast search and the result of TSCSO increased by 2.4%, 6.9%, and 7.7% compared with the highest accuracy in extractive models. Also, to increase accuracy by 8.3%, 11%, and 10.6% compared with the lowest accuracy in extractive models but may be stuck in a local optimum so genetic algorithm solved this problem. We validated the proposed approach on the popular summarization dataset and observed consistent improvement over baseline models. The CSOGA algorithm achieved the best f-measure but need much time to reach the best score. According to the results, it is clear that the TSCSO algorithm is faster to reach a solution and has more stability. For future work, study new ways to improve stability and speed of the proposed hybrid optimization algorithm, also, other swarm algorithms can be explored to try to make other more powerful hybrid algorithms for text summarization.

Conflicts of Interest

The authors declare that we do not have any circumstances or interests that may affect the results discussed in this manuscript.

Author Contributions

Conceptualization, Mostafa Gamal;

methodology, Ahmed El-Sawy; software, Mostafa Gamal; validation, Ahmed H. AbuEl-Atta, and Ahmed El-Sawy; formal analysis, Mostafa Gamal and Ahmed H. AbuEl-Atta; investigation, Ahmed El-Sawy;

Resources, Mostafa Gamal; data curation, Mostafa Gamal; writing — Mostafa Gamal, Ahmed H. AbuEl-Atta; writing — review and editing, Ahmed El-Sawy.

References

[1] K. Jezek and J. Steinberger, "Automatic Text Summarization: the state of the art 2007 and new challenges", In*: Proc. of Znalosti*, pp. 1-12, 2008.

[2] E. H. Hovy, "Automated text summarization", R. Mitkov (Ed.), *The oxford handbook of computational linguistics, Oxford: Oxford University Press*, pp. 583–598, 2005.

[3] A. Al-Saleh and M. E. B. Menai, "Solving multi-document summarization as an orienteering problem", *Algorithms*, Vol. 11, No. 7, p. 96, 2018.

[4] L. M. Al Qassem, D. Wang, Z. Al Mahmoud, H. Barada, A. Al-Rubaie, and N.I. Almoosa, "Automatic Arabic Summarization: A survey of methodologies and systems", *Procedia Computer Science*, Vol. 117, pp. 10-18, 2017.

[5] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using latent semantic analysis", *Journal of Information Science*, Vol. 37, No. 4, pp. 405-417, 2011.

[6] I. Imam, N. Nounou, A. Hamouda, H. Allah, and A. Khalek, "Query Based Arabic Text Summarization", *IJCST*, Vol. 1, pp. 2, 2013.

[7] R. Al-Hashemi, "Text Summarization Extraction System (TSES) Using Extracted Keywords", *Int.*

Arab J. e-Technol, Vol. 1, No. 4, pp. 164-168, 2010.

[8] 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.

[9] V. V. M. K. Ravinuthala and S. R. Chinnam, “A Keyword Extraction Approach for Single Document Extractive Summarization Based on Topic Centrality”, *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 5, pp. 153-161, 2017.

[10] A. Aries, D. E. Zegour, and K. W. Hidouci, “All Summarizer system at MultiLing 2015: Multilingual single and multi-document summarization”, In: *Proc. of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 237-244, 2015.

[11] A. Sinha, A. Yadav, and A. Gahlot, “Extractive text summarization using neural networks”, *arXiv* preprint arXiv: 1802.10137. 2018.

[12] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing”, *IEEE Computational Intelligence Magazine*, Vol. 13, No. 3, pp. 55-75, 2018.

[13] M. El-Gedawy, “Comparing pmi-based to cluster-based arabic single document summarization approaches”, *International Journal of Engineering Trends and Technology (IJETT)*, Vol. 11, pp. 379-383, 2014.

[14] A. T. Al-Taani and M. M. Al-Omour, “An extractive graph-based Arabic text summarization approach”, *The International Arab Conference on Information Technology*, 2014.

[15] S. Thomas, C. Beutenmüller, X. de la Puente, R. Remus, and S. Bordag, “Exb text summarizer”, In: *Proc. of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 260-269, 2015.

[16] R. Mihalcea, and P. Tarau, “Textrank: Bringing order into text”, In: *Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404-411, 2004.

[17] S. Lagrini, M. Redjimi, and N. Azizi, “Automatic arabic text summarization approaches”, *International Journal of Computer Applications*, Vol. 164, No. 5, pp. 31-37, 2017.

[18] S. Mandal, A. Singh, “Text Summarization Technique by Sentiment Analysis and Cuckoo Search Algorithm”, *Computing in Engineering and Technology, Advances in Intelligent Systems and Computing*, Springer, Singapore, Vol. 1025, 2020.

[19] 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.

[20] Q. A. Al-Radaideh and D. Q. Bataineh, “A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms”, *Cognitive Computation*, Vol. 10, No. 4, pp. 651-669, 2018.

[21] H. P. Edmundson, “New methods in automatic extracting”, *Journal of the ACM (JACM)*, Vol. 16, No. 2, pp. 264-285, 1969.

[22] S. Sekine and C. Nobata, “Sentence extraction with information extraction technique”, In: *Proc. of the Document Understanding Conference*, 2001.

[23] M. A. Fattah and F. Ren, “Automatic text summarization”, *World Academy of Science, Engineering and Technology*, Vol. 37, 2008.

[24] W. Luo, F. Zhuang, Q. He, and Z. Shi, “Effectively Leveraging Entropy and Relevance for Summarization”, *Information Retrieval Technology: 6th Asia Information*, 2010.

[25] M. Cooper and J. Foote, “Summarizing video using non-negative similarity matrix factorization”, *IEEE Workshop on Multimedia Signal Processing*, pp. 25-28, 2002.

[26] V. Thada and D.V. Jaglan, “Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm”, *International Journal of Innovations in Engineering and Technology (IJIET)*, Vol. 2, No. 4, pp. 202, 2013.

[27] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm”, *Applied soft computing*, Vol. 8, No. 1, pp. 687-697, 2008.

[28] X. Meng, Y. Liu, X. Gao, and H. Zhang, “A new bio-inspired algorithm: chicken swarm optimization”, *International Conference in Swarm Intelligence*, Springer, Cham, pp. 86-94, 2014.

[29] N. Niazy, A. El-Sawy, and M. Gadallah, “A Hybrid Chicken Swarm Optimization with Tabu Search Algorithm for Solving Capacitated Vehicle Routing Problem”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 237- 247, 2020.

[30] R. Paulus, C. Xiong, and R. Socher, “A deep reinforced model for abstractive summarization”, In: *Proc. of the 6th*

International Conference on Learning Representations, Vancouver, Canada, 2018.

[31] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization", In: *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109, 2018.

[32] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep communicating agents for abstractive summarization", In: *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 1662–1675, 2018.

[33] M. Zhong, P. Liu, D. Wang, X. Qiu, and X. Huang, "Searching for effective neural extractive summarization: What works and whats next", In: *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1049–1058, 2019a.

[34] L. Luo, X. Ao, Y. Song, F. Pan, M. Yang, and Q. He, "Reading like her: Human reading inspired extractive summarization", In: *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3024–3034, 2019.

[35] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning", In: *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 1747–1759, 2018.

[36] X. Zhang, M. Lapata, F. Wei, and M. Zhou, "Neural latent extractive document summarization", In: *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 779–784, 2018.

[37] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences", In: *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 654–663, 2018.

[38] Y. Liu, I. Titov, and M. Lapata, "Single document summarization as tree induction", In: *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 1745–1755, 2019.

[39] A. See, P.J. Liu, and C.D. Manning, "Get to the point: Summarization with pointer generator networks", In: *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, pp. 1073– 1083, 2017.