

A Hybrid PSO-GA for Extractive Text Summarization

Bui Thi Mai Anh, Nguyen Thi Thu Trang*, Tran Thi Dinh, Nguyen Thi Hoai

Hanoi University of Science and Technology

School of Information and Communication Technology

anhbhm@soict.hust.edu.vn, trangntt@soict.hust.edu.vn
dinh.tt173015@sis.hust.edu.vn, hoai.nt202195m@sis.hust.edu.vn

Abstract

In extractive summarization, searching for the most salient sentences to generate summaries can be resolved by applying meta-heuristic optimization algorithms including genetic algorithms (GAs), particle swarm optimization (PSO), etc. In these approaches, the best solution, i.e. the expected summary, is found by optimizing an objective function, taking into account the features of document and/or sentence. However, these traditional algorithms alone may suffer from a weak local search capability and slow convergence speed. To this end, this paper proposes a novel hybrid GA-PSO algorithm, namely PSOGA-BKSum, for single document text summarization based on a scoring feature subset and updating the candidate toward the best solution technique. The experiment results on the two common extractive datasets, DUC2001 and DUC 2002, have shown that PSOGA-BKSum outperforms some state-of-the-art works on all three ROUGE point metrics. PSOGA-BKSum has considerably progressed on DUC2001 dataset, from 4.3 to 9.7% higher (i.e. 8.5 to 21.9% relatively) than the best ROUGE-1 score of previous un-

supervised algorithms for this problem. For the dataset DUC2002, the PSOGA-BKSum receives the higher points on ROUGE-1 and ROUGE-2 scores, around 4.3% to 8.7% than the best previous algorithms.

1 Introduction

The automatic text summarization allows users to reach information in a shorter form without losing any important aspects presented therein. This task becomes increasingly important due to the explosion of textual information on the Internet. Depending on the way in which summaries are generated, there are two approaches of text summarization (Hahn and Mani, 2000). Extractive summaries are formed by selecting from the original document the most salient sentences. Abstractive summaries, on the other hand, require advanced linguistic techniques to rephrase and generate new sentences which are not in the original document, therefore are much more complex than the former.

The extractive approach can be studied as a binary classification task in which sentences from the input document are split into two groups: *in-summary* and *not-in-summary*. There is a huge diversity of techniques to automatically generate extractive summaries for a

*Corresponding author

single document which can be grouped into two directions: *supervised* and *unsupervised*. The former methods are based on machine learning models (Wong et al., 2008) or deep learning models (Liu, 2019), those require a huge training dataset including human-generated summaries, then very costly and time consuming. The latter approaches, on the other hand, do not require any training corpus but aim to rank all the sentences from the original document by exploring their relationship. The sentences with the highest ranking scores are then selected to build the summary. In this research direction, many techniques have been proposed for extractive summarization such as Hidden Markov Model (Yang et al., 2014), graph-based model (Mihalcea, 2004), algebraic reduction (Batcha and Zaki, 2010) and evolutionary algorithms for instance, Genetic Algorithm (GA) (Mendoza et al., 2014; Anh et al., 2019), Particle Swarm Optimization (PSO) (Foong and Oxley, 2011).

Evolutionary algorithms (EAs) have shown promising results in solving the problem of extractive summarization. In this context, EAs are cast as a clustering method that aim to search for the most relevant sentences from the input document. The search process is guided by fitness functions who assign a value to each candidate solution (i.e., candidate summary) in the search space to assess its quality. Most studies adopted Genetic Algorithms (GAs) and focused on exploring the effectiveness of different strategies to formalize fitness functions (Meena and Gopalani, 2015; Anh et al., 2019). The essential advantage of GAs is the capacity to maintain the diversity of candidate solutions thanks to genetic operators (i.e., selection, crossover and mutation). However, the complexity of the evolution process often leads to a slow convergence speed. Indeed, the

computation time of GA increases non-linearly in the case of large population size. In this paper, we propose a hybrid PSO-GA algorithm for extractive summarization. On one hand, the low speed limitation of GA is overcome by combining with PSO - a fast convergent searching algorithm. On the other hand, the local optimum phenomenon which is considered as a drawback of PSO is avoided thanks to GA. The proposed algorithm can benefit and exploit the advantages of both PSO and GA approaches.

The rest of the paper is organized as follows. Section 2 presents in detail our proposal with the hybrid PSO-GA for extractive summarization problem. Section 3 discusses the related works to the model and techniques in our model. In section 4, the experiments and evaluations on DUC2001 and DUC2002 are described and discussed. Section 5 shows and discusses on experimental results of our proposed method with other state-of-the-art methods. Finally, the paper draws some conclusions and perspectives of the work in Section 6.

2 Proposal of PSOGA-BKSum for Extractive Summarization

Genetic algorithm (GA) (Theede, 2004) imitates the natural evolutionary process to solve optimization problems. In the context of GA, candidate solutions are *individuals* which are characterized by their *chromosome*. Whereas, Particle Swarm Optimization (PSO), first proposed by Eberhart and Kennedy (Eberhart and Kennedy, 1995), simulates collaborative activities within a swarm of herds (e.g., a flock of birds) in order to search for the best food resource. Comparing to GA, the mechanism to produce a new population of solutions based on the floating point arithmetic allows PSO to converge rapidly. However, because of the high

density of population in the solution space, PSO tends to be trapped in local optimum.

In this paper, we introduce a novel hybrid algorithm for extractive summarization, namely PSO-GA, to exploit the advantages of both PSO and GA approaches. The drawback of PSO is overcome by applying GA to achieve the diversification of population, which may helps to overcome the local optimum phenomenon. Besides, the low speed limitation of GA is overcome by combining with PSO - a fast convergent searching algorithm.

2.1 Problem Representation

Given a document $\mathcal{D} = \{S_1, S_2, \dots, S_N\}$ where S_i denotes the i -th sentence of \mathcal{D} . In the context of PSO, candidate summaries are represented by particles in the N -dimensional search space. For each particle, the position is encoded by a binary vector of N elements, $X = \{0, 1, 0, \dots, 0, 0\}$ where 1s/0s denote that the corresponding sentences are included/not included in the summary, respectively. The velocity is a real vector of N elements in the range $[0, 1]$ allowing to redirect particles towards the best solution. Since the position space is discrete (i.e., binary), a rounding operation is needed to update particles' velocity and position as follows:

$$x_j^i = \begin{cases} 1 & \text{if } x_j^i \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where x_j^i denotes the j -th element of the position vector X_i .

2.2 Hybrid PSO-GA Algorithm

The proposed algorithm is depicted in Algorithm 1. The input of the algorithm includes the input document of N sentences together with the parameters of PSO and GA (including the population size $popSize$, number of iterations

of PSO/GA, ω , c_p , c_g and genetic operators' parameters). The first stage of the process is to generate randomly the initial population for PSO (line 1). For each generated particle's position, the number of elements 1 is checked to satisfy the constraint on summary's length (Equation 2).

$$\sum_{j=1}^N x_j = \ell \quad (2)$$

Algorithm 1: The proposed PSOGA-BKSum for extractive summarization

Input : Document D with N sentences

Output: Summary of ℓ sentences

```

1  $\mathcal{P}_{PSO} \leftarrow initializePop(popSize)$ 
2  $k \leftarrow 1$ 
3 while  $k < nbIteration_{PSO}$  do
4   foreach  $p \in \mathcal{P}_{PSO}$  do
5     compute  $\mathcal{F}(p)$ 
6     compute  $pbest_p, gbest$ 
7     update  $X_p, V_p$ 
8   end
9    $Sorting(\mathcal{P}_{PSO})$ 
10   $\mathcal{P}_{GA} \leftarrow \mathcal{P}_{PSO}[1 : \frac{popSize}{2}]$ 
11   $\mathcal{P}_{PSO} \leftarrow \mathcal{P}_{GA}$ 
12   $j \leftarrow 1$ 
13  while  $j < nbIteration_{GA}$  do
14     $\mathcal{P}_{GA} \leftarrow select(\mathcal{P}_{GA})$ 
15     $crossOver(\mathcal{P}_{GA})$ 
16     $mutate(\mathcal{P}_{GA})$ 
17    compute  $\mathcal{F}(p) \forall p \in \mathcal{P}_{GA}$ 
18     $j \leftarrow j + 1$ 
19  end
20   $\mathcal{P}_{PSO} \leftarrow \mathcal{P}_{PSO} + \mathcal{P}_{GA}$ 
21   $Evaluation-Convergence(\mathcal{P}_{PSO})$ 
22   $i \leftarrow i + 1$ 
23 end
24 return The best summary of  $D$ 
```

The quality of all particles is assessed using

the fitness function \mathcal{F} . According to their fitness value, the personal best and the global best position are determined (i.e., $pbest_p$, $gbest$) then the position and velocity of all particles are updated using Equation ?? (lines 4-8). In order to avoid the high density of solutions in the solution space of PSO, we apply a genetic algorithm on a half of PSO population (lines 13-19). The genetic operators of GA (lines 15, 16 and 17) allow to diversify the original population of PSO, therefore prevent the premature convergence. These operators and the fitness function will be detailed in the rest of this section.

2.3 Fitness Function

Fitness functions play an essential role in optimization searching algorithms to qualify solutions. In this study, we apply the same fitness function for the proposed PSO and GA algorithms. In order to assess the quality of candidate summaries, our fitness function is built based on four sentence features including sentence position, similarity to the topic sentence, sentence length and number of proper nouns. Sentences from the original document are first represented in the Vector Space Model using Term Frequency and Inverse Document Frequency (TF-IDF) (Manning et al., 2008). As such, with a given document \mathcal{D} containing \mathbb{M} terms, each sentence S_i is represented by a weighing vector:

$$S_i = \{w_{i1}, w_{i2}, \dots, w_{ik}, \dots, w_{i\mathbb{M}}\}$$

where w_{ik} denotes the weight of term t_k in the i -th sentence. The formula of this weight is depicted in Equation 3.

$$w_{ik} = (f_{ik}/MaxFreq_i) \times \log\left(\frac{N}{n_k}\right) \quad (3)$$

where f_{ik} is the frequency of term t_k in the sentence S_i , $MaxFreq_i$ is the maximum frequency through measuring all terms in S_i . In

the original formula of TF-IDF, n_k is the number of documents containing the term t_k . We adapt this formula to our work by calculating n_k as the number of sentences in the document which include t_k . N is the total number of sentences in \mathcal{D} .

2.3.1 Sentence Position

Recent studies have shown that the most important information tends to be appeared in special parts of a document such as titles, headings, the opening of paragraphs etc. (Mendoza et al., 2014). Assessing a summary based on the position of its sentences allow to determine the difference in terms of distance between them and the key sentences of the document. The position feature of the candidate summary \mathcal{S} is evaluated as follows:

$$\mathcal{P}_{\mathcal{S}} = \frac{\sum_{\forall S_i \in \mathcal{S}} N - Pos(S_i)}{N} \quad (4)$$

where $Pos(i)$ refers to the position of the i -th sentence S_i in the original document. In this sense, $\mathcal{P}_{\mathcal{S}}$ tends to prefer summaries which contain the first sentences of the document.

2.3.2 Similarity to the topic sentence

The topic sentence provides the key information of a document. Sentences of a summary should relate to the topic sentence to show the relevant content of the document. In order to evaluate the relevance of summary's sentences, we calculate the *cosine similarity* between them and the topic sentence which is typically the head sentence of the document. The cosine similarity formula between two sentences is depicted in Equation 5.

$$sim(S_i, S_j) = \frac{\sum_{k=1}^M (w_{ik} \times w_{jk})}{\sqrt{\sum_{k=1}^M (w_{ik}^2) \times \sum_{k=1}^M (w_{jk}^2)}} \quad (5)$$

where w_{ik}, w_{jk} denotes the weight of the term k in the sentence S_i and S_j respectively. The similarity to the topic sentence of the candidate summary \mathcal{S} is calculated as in Equation 6.

$$\mathcal{R}_S = \frac{\sum_{\forall S_i \in \mathcal{S}} \text{sim}(S_i, S^*)}{\ell} \quad (6)$$

Where S^* is the topic sentence and ℓ is the prefixed length of summaries (in sentences).

2.3.3 Sentence Length

Very short sentences are less likely to appear in the summary as they contains less information (Meena and Gopalani, 2015). Given a sentence S_i in the summary \mathcal{S} ($x_i \neq 0$), the length in terms of number of words of S_i is calculated and normalized as follows:

$$\hat{l}_i = \frac{l(S_i) - \mu}{\sigma}$$

where $l(S_i)$ is the length of the i -th sentence, μ and σ denote the average and standard deviation calculated from the length of all sentences in the summary. The sentence length feature of the summary \mathcal{S} is measured as in Equation 7.

$$\mathcal{L}_S = \frac{\sum_{\forall x_i=1 \in \mathcal{S}} \hat{l}_i}{\max_{\forall S_i \in \mathcal{S}} \hat{l}_i} \quad (7)$$

where $\max \hat{l}_i$ denotes the length of the longest sentence in the summary.

2.3.4 Number of Proper Nouns

Proper nouns often provide important information and characterize the original document. We therefore calculate the number of proper nouns in a candidate summary to measure this characteristic. The proper noun factor of a summary can be calculated as in Equation 8.

$$\mathcal{N}_S = \frac{\sum_{\forall S_i \in \mathcal{S}} \mathcal{N}_{S_i}}{\mathcal{N}_{\mathcal{D}}} \quad (8)$$

where \mathcal{N}_{S_i} represents the number of proper nouns of the sentence S_i , $\mathcal{N}_{\mathcal{D}}$ refers to the total number of proper nouns of the document \mathcal{D} .

The fitness function \mathcal{F} is formalized as a linear combination of all mentioned-above features as indicated in Equation 9.

$$\mathcal{F}(\mathcal{S}) = \alpha \hat{\mathcal{R}}_S + \beta \mathcal{L}_S + \gamma \mathcal{P}_S + \sigma \mathcal{N}_S \quad (9)$$

where $\alpha, \beta, \gamma, \sigma$ are coefficients to describe the contribution of each feature, which satisfy Equation 10.

$$\alpha + \beta + \gamma + \sigma = 1 \quad (10)$$

As each particle in the current population is characterized by a binary vector representing whether or not a corresponding sentence belongs to the summary or not. To evaluate a particle, we calculate all these aforementioned features based on its structure. The coefficients α, γ, β and σ will be determined through parameter turning phase (see Section 4.2).

2.4 Genetic Operators

The reproduction process of GA is achieved by repeated applying three operations: selection, crossover and mutation.

2.4.1 Selection

The better candidates are selected from the current population to become parents. The probability of selection is typically defined using the relative ranking of fitness values. In this study, we adopt the Roulette Wheel selection strategy (Thede, 2004). As indicated in Equation 11, the contribution of each individual to the total fitness will decide whether or not this individual is selected.

$$p_i = \frac{\sum_{j=1}^i f_j}{\sum_{k=1}^{\mathbf{P}} f_k} \quad (11)$$

where f_i is the fitness value of the i -th individual and P is the size of the population.

2.4.2 Crossover

The crossover operator combines two parents to produce new offspring. We consider the *uniform crossover* strategy in which each gen from either parent has an equal probability to be chosen (Thede, 2004). Given two parents S_f and S_m , the new offspring is built as following:

$$S_c[i] = \begin{cases} S_f[i] & (p_i \geq a_{cross}) \\ S_m[i] & \text{otherwise} \end{cases} \quad (12)$$

where p_i is randomly picked in the range $[0..1]$ for each gen and a_{cross} is a random mixing ratio ($a_{cross} \in [0..1]$). We exchange the role of parents to generate the second offspring with the same value of p_i . The constraint on summary's length should be verified for new offspring to assure that the number of elements 1 does not exceed ℓ (see Equation 2).

2.4.3 Mutation

There might be a tendency that new solutions become similar after several iterations because fitter individuals are more likely to be chosen. Mutation operators are then applied to assure the diversity of population. We consider that each gen of a given chromosome has an equal probability to be mutated. The mutation operator is described in Equation 13

$$S_c[i] = \begin{cases} 1 & (p_i \geq a_{mut}) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where p_i is also randomly picked in the range $[0..1]$ and a_{mut} is the probability of mutation. Before mutating, the constraint described in Equation 2 is verified. If the restriction is not met, the mutation is declined.

2.4.4 Coverage Evaluation

Once GA completes, we apply a convergence evaluation at the end of PSO-iteration to avoid population stagnation. We define the convergence as follows.

$$Conv = \begin{cases} True & \text{if } |\mathcal{E}_v| \geq 0.9 * popSize \\ False & \text{otherwise} \end{cases} \quad (14)$$

where $\mathcal{E}_v = \{p | \frac{F(p)}{\mu_F} \in [0.9..1.1]\}$ includes particles in the current population whose fitness values are in the threshold of 10% around the average fitness μ_F , $|\mathcal{E}_v|$ denotes the size of \mathcal{E}_v . If the convergence condition is reached, we re-initialize the PSO population.

3 Related Works

Evolutionary algorithms have been applied in many disciplines and have shown promising results for extractive summarization task. Many heuristic search algorithms have been proposed in which Genetic Algorithm, Particle Swarm Optimization, Differential Evolutionary are the most attractive.

Genetic Algorithm (GA). Most studies applied GA to tackle the problem of extractive summarization. Mendoza et al. proposed a GA approach in which they combined genetic operators with local search heuristic (Mendoza et al., 2014). Their proposed model, naming MA-SingleDocSum, takes advantage of the evolutionary process of GA as well as a guided local search strategy to explore the sentence search space of the document. Other works aimed at applying GAs to explore the effectiveness of sentence features to construct the fitness function (Meena and Gopalani, 2015). The studies showed that some sentence features including proper noun, sentence position and named entities contribute the most to the quality of

summaries. The recent research of Bui et al. provided an analysis on different crossover and mutation strategies (Anh et al., 2019). They also proposed a guided mutation strategy to achieve the better performance when generating new candidate solutions.

Particle Swarm Optimization (PSO). Comparing to GA, the evolutionary process of PSO is more simple, leading to a significant computational advantage over GA. Binwahlan et al. early applied PSO to investigate the contribution of different features to qualify generated summaries (Binwahlan et al., 2009). They however targeted to solve the feature selection problem. Raed and Ahmad proposed another approach to combine informative and semantic features with PSO algorithm to extract summaries from Arabic documents (Al-Abdallah and Al-Taani, 2017).

Other heuristic algorithms. Differential Evolution (DE) has been also studied for extractive summarization. Aliguliyev proposed a clustering approach in which the author measure the dissimilarity between sentences based on normalized google distance (Aliguliyev, 2009). A DE algorithm was then applied to optimize the binary clustering of sentences. Another work of Foong and Oxley (Foong and Oxley, 2011) focused on a combination of PSO and Harmony Search for generating extractive summaries. In this study, Harmony Search algorithm was used to store feature weights obtained from PSO process and to evaluate the importance of these features.

4 Empirical Settings

4.1 Dataset

For the text summarization task, there exists several datasets including Daily-

Mail/CNN (Hermann et al., 2015), and Document Understanding Conference (DUC) (Over and Liggett, 2002) with different versions such as DUC2001, DUC2002, DUC2004 etc. As far as we know, DailyMail/CNN provides only abstractive summaries from news and articles. The DUC2004 meanwhile only includes summaries for multi-document summarization task. We therefore did experiments on DUC2001 and DUC2002 to assess our proposed approach as they have been widely adopted for evaluate the single document extractive summarization.

The DUC2001 dataset includes 309 documents which are categorized into 30 sets while the DUC2002 dataset includes 59 sets of 567 documents retrieved from news reports in English. Each set from both datasets provides reference 100-word summaries for single and multiple documents.

4.2 Metrics and Parameter Tuning

Common metrics for evaluating automatic text summarization are two types of ROUGE scores (Lin, 2004): ROUGE-N and ROUGE-L. This measure is computed by counting the number of overlapping words between the ground truth summary and generated summaries. ROUGE-N can be thought of as the overlapping of N-grams where N takes the value of 1 and 2. Meanwhile, Rouge L (Longest Common Subsequence), measures the longest matching sequence of words based statistics. In this paper, we also adopt ROUGE-1,2 and ROUGE-L to assess the performance of our proposed approach.

Table 1 shows the values of PSO-GA parameters which are used in our experiments. These parameters and feature weights are empirically determined by applying a grid search on training data. To create a disjoint training and test data, we divide each data-set into equally sized

Table 1: PSO-GA parameters setting

Parameter	Value
Number of iterations PSO I_{PSO}	50
Number of iterations GA I_{GA}	20
Population Size $popSize$	30
Inertia factor ω	0.5
Personal Acceleration Coefficient c_p	0.5
Social Acceleration Coefficient c_g	0.9
Mutation Rate Mr	0.4

folds: three folds for DUC2001 and five folds for DUC2002. We evaluate parameters on $fold_i$ to maximize the ROUGE scores and tested on $fold_{i+1}$. Finally, we collect the results from all test folds and compute the overall performance of the approach.

4.3 Experimental algorithms

To evaluate the performance of the proposed algorithm, we did some experiments with other state-of-the-art works on extractive single-document summarization as follows:

- GA-Features (Anh et al., 2019): A GA-based model was proposed to evaluate the effectiveness of features and genetic operators to obtain qualified summaries.
- MA-SingleDocSum (Mendoza et al., 2014): The authors combined GA and a guided local search module to extract summaries.

Beside the proposed hybrid PSO-GA, we also run the experiment on our only GA or PSO algorithm to see the effectiveness of the hybrid model. As a result, the experiment was conducted on:

- PSOGA-BKSum: Our proposed approach

which combines PSO and GA for extractive summarization.

- GA-BKSum: Our proposed GA in PSOGA-BKSUM for extractive summarization.
- PSO-BKSum: Our proposed PSO for extractive summarization.

5 Results

Table 2 and Table 3 show the performance of all these aforementioned models carried out on the dataset DUC2001 and DUC2002. In this experimentation, we also examined two algorithms PSO and GA separately (i.e., GA-BKSum and PSO-BKSum). It can be observed that our proposed approach, PSOGA-BKSum, consistently outperforms the six baseline methods in terms of ROUGE-1,2 and ROUGE-L. For example, comparing to the recent method MA-SingleDocSum on the dataset DUC2001, PSOGA-BKSum increased ROUGE-1 by an average of 9.68%, the improvements at ROUGE-2 and ROUGE-L are 6.11% and 13.25% (comparing to MA-SingleDocSum reimplementation), respectively. For the dataset DUC2002, our method generally achieved an increase from 4.98% to 8.71% on ROUGE-1, from 4.25% to 4.66% on ROUGE-2 and from 7.77% to 10.41% on ROUGE-L.

We also experimented separately the proposed PSO (i.e., PSO-BKSum) and GA (i.e., GA-BKSum). It is reasonable that the approach based on GA obtained a better performance than PSO. Indeed, GA-BKSum has the ROUGE-1 score 52.78%, about 1.76% higher than PSO-BKSum.

Comparing to GA-Features (Anh et al., 2019), our proposed model with only GA, GA-BKSum, has increased the ROUGE-1 score

Table 2: Comparing the ROUGE Scores on DUC 2001 Dataset (%)

Method	ROUGE Scores		
	ROUGE-1	ROUGE-2	ROUGE-L
GA-Features (re-run)	50.26	21.63	42.10
MA-SingleDocSum (published)	44.86	20.14	-
MA-SingleDocSum (re-implement)	44.17	20.68	38.67
<i>GA-BKSum (our but only GA)</i>	52.78	24.39	48.21
<i>PSO-BKSum (our but only PSO)</i>	47.31	20.28	43.12
PSOGA-BKSum (our hybrid)	54.54	26.25	49.92

Table 3: Comparing the ROUGE Scores on DUC 2002 Dataset (%)

Method	ROUGE Scores		
	ROUGE-1	ROUGE-2	ROUGE-L
GA-Features (re-run)	52.01	23.25	44.57
MA-SingleDocSum (published)	48.28	22.84	-
MA-SingleDocSum (re-implement)	48.33	20.60	41.93
<i>GA-BKSum (our but only GA)</i>	54.83	26.11	51.11
<i>PSO-BKSum (our but only PSO)</i>	51.17	23.18	46.67
PSOGA-BKSum (our hybrid)	56.99	27.50	52.34

by 2.52% on DUC2001 and by 2.82% on DUC2002. The improvements at ROUGE-2 and ROUGE-L are 2.76% and 6.11% on DUC2001, 2.86% and 6.54% on DUC2002. On the one hand, we used the different features which seem to be more effective than those of GA-Features. On the other hand, we applied a strategy to re-initialize the population to improve the performance of GA. However, the full approach allows to achieve the best performance.

6 Conclusion

In this paper, we have introduced a novel hybrid PSO-GA algorithm for solving the extractive summarization problem. Our proposed algorithm aims to speed up the traditional GA by taking advantage of PSO. The premature convergence of PSO is prevented by applying GA

on a small population. As such, the advantages of both approaches are benefited. Our proposal is experimented to compare with state-of-the-art methods using ROUGE measures on the datasets DUC2001 and DUC2002, those are commonly used for single document text summarization task. The empirical results have shown that the proposed hybrid PSO-GA gives better accuracy than other state-of-the-art methods of the same direction research. The application of evolutionary algorithms for solving the multi-document summarization is currently trending. Extending our proposed algorithm to target to multi-document extractive summarization is therefore our future direction.

References

- Raed Z Al-Abdallah and Ahmad T Al-Taani. 2017. Arabic single-document text summarization using particle swarm optimization algorithm. *Procedia Computer Science*, 117:30–37.
- Ramiz M Aliguliyev. 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, 36(4):7764–7772.
- Bui Thi Mai Anh, Nguyen Tra My, and Nguyen Thi Thu Trang. 2019. Enhanced genetic algorithm for single document extractive summarization. In *Proceedings of the Tenth International Symposium on Information and Communication Technology*, pages 370–376.
- Nowshath Kadhar Batcha and Ahmed M Zaki. 2010. Algebraic reduction in automatic text summarization—the state of the art. In *International Conference on Computer and Communication Engineering (ICCCE’10)*, pages 1–6. IEEE.
- Mohammed Salem Binwahlan, Naomie Salim, and Ladda Suanmali. 2009. Swarm based features selection for text summarization. *IJCSNS International Journal of Computer Science and Network Security*, 9(1):175–179.
- Russell Eberhart and James Kennedy. 1995. A new optimizer using particle swarm theory. In *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43. Ieee.
- Oi-Mean Foong and Alan Oxley. 2011. A hybrid pso model in extractive text summarizer. In *2011 IEEE Symposium on Computers & Informatics*, pages 130–134. IEEE.
- Udo Hahn and Inderjeet Mani. 2000. The challenges of automatic summarization. *Computer*, 33(11):29–36.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.
- Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Cambridge university press.
- Yogesh Kumar Meena and Dinesh Gopalani. 2015. Evolutionary algorithms for extractive automatic text summarization. *Procedia Computer Science*, 48:244–249.
- Martha Mendoza, Susana Bonilla, Clara Noguera, Carlos Cobos, and Elizabeth León. 2014. Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications*, 41(9):4158–4169.
- Rada Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 170–173.
- Paul Over and Walter Liggett. 2002. Introduction to duc: an intrinsic evaluation of generic news text summarization systems. *Proc. DUC*. <http://www.nlpir.nist.gov/projects/duc/guidelines/2002.html>.
- Scott M Thede. 2004. An introduction to genetic algorithms. *Journal of Computing Sciences in Colleges*, 20(1):115–123.
- Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 985–992. Association for Computational Linguistics.
- Libin Yang, Xiaoyan Cai, Yang Zhang, and Peng Shi. 2014. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information sciences*, 260:37–50.