Optimization-Based Techniques for Sentiment-Oriented Text Summarization: A Concise Review

Abeer Raad[1*], Rafid Sagban[2]

Abstract

In natural language processing (NLP), text summarization is a process of converting a big textual information from single or multi documents into a concise text without change its semantics. The variety of summarization procedures in literature leads to different processes have their own pros and cons. Text summarization expressed as an NP-complete problem. Thus, optimization-based summarization methodologies is the only available framework for solving such kind of mathematical problems in A.I. This paper reviews algorithms that express the problem in such a way that optimize text summarization to get high accuracy.

Key Words: document summarization, optimization-based, Genetic Algorithm, Particle Swarm Optimization, Ant colony optimization, Artificial Bee colony

DOI Number: 10.14704/nq.2022.20.8.NQ44245 **NeuroQuantology 2022; 20(8):2230-2238**

Introduction

The demand for document summarizing technique is growing because of a rapid increase in the text data alongside for the use of digital technology as a means of extract information from raw data and spread it in cyberspace. Artificial Intelligence techniques designed for automatic document summarizing. There are three major steps for automatic document summarization: pre-processing, extract the features, and represent the summary. To summarize, several methods were used, such as, topic-based, discourse-based, graph-based, statistical-based, and Machine Learning systems [1]. Meta-heuristics optimization based methods used to accomplish these approaches. It is one of the methods of artificial intelligence based on stochastic to find the optimal solution in each iteration[2]. This method is based on individual solutions and population-based solutions to find the global (near-) optimal solution [3]. It depends on finding the next optimal solution on the previous experiences of individuals and on the other experiences of the rest of the individuals in population. Meta-heuristic optimization methods have become one of the most promising methods in the last twenty years for solving complex combinatorial problems. It explores large search spaces to find local and global solutions in order to implement optimization. Because of the way this meta-heuristic technique works, it can find a global optimum solution[4]. Swarm Intelligence (SI) and Evolutionary Algorithms (EA) are two methodologies based on the concept of meta-heuristic. As they searched optimal or approximate solutions in a large search space to solve complex combinational problems and avoid falling into the trap of local solutions [5]. The SI and EA area of research are rich problem solving frameworks in optimization.

The process of optimization is restricted in several steps such as the initialization of population, the iterative process of optimization, the evaluation of objective function and the stop condition. The population is initialized with some solutions selected randomly, or by simple heuristics such dispatch rules.

Corresponding author: Abeer Raad
Address: [1]Department of Software, Information Technology College University of Babylon Hilla, Iraq, [2]Department of Computer Technology Engineering, Technical Engineering College, Al-Ayen University, Thi-Qar, Iraq
E-mail: abeerraad.sw.msc@student.uobabylon.edu.iq[1], rsagban@alayen.edu.iq[2]

All decision variables' values should be inside their prescribed ranges or domains. The produced solutions then evaluated by the number of pre-determined objective functions. An iteration process will find new solutions using various techniques. New solutions assessed and replaced in the population based on a set of pre-prepared rules. The associated solutions' objective values updated. The iteration procedure repeated until met the stop condition. Finally, the best solution is being selected, together with the related objective outcomes [6]. The mechanisms for generating new solutions to a variety SI and EA algorithms differ. In recent years, a growing number of Swarm Intelligence (SI) and Evolutionary Algorithms (EA) have been used to solve automatic text summarization problems, including the Genetic Algorithm (GA)[7][8][9][10][11][12][13], Particle Swarm Optimization (PSO)[14][15], Ant Colony Optimization (ACO)[16][17][18], Artificial Bee Colony (ABC)[19][20][21] , and others.
This study presents the literature of optimization algorithms to generate automatic document summarization. According to the findings of reference searches and empirical research, Particle Swarm Optimization, Genetic Algorithms, Ant Colony Optimization, And Artificial Bee Colony Optimization are optimization algorithms that have superior performance in constructing automatic summarization. After exposure to the optimization algorithm and summary steps. It continues with a discussion of the findings of the gap analysis of the optimization algorithms as automatic document summarization methods. The overall conclusions presented at the end of the paper.

Text summarization

The necessity for automatic text summary (ATS) has arisen because of the increase in online publishing, enormous numbers of internet users, and the rapid expansion of electronic governance (e-government). Due to the rapid development of information communication technologies, a vast number of electronic documents are now available on the internet, and users are having difficulty finding relevant information. Furthermore, the internet has made vast collections of text on a wide range of topics available. This explains why there is so much redundancy in the online texts. Users become so fatigued after reading a huge number of texts that they may overlook many important and relevant documents. As a result, in this phase, a reliable text summarizing system is required. These systems can condense information from a variety of publications into a concise, understandable summary [22][23] . Four main goals were considered by [24]: information coverage, information significance, information redundancy, and text cohesion.
There are two primary approaches to text summarization: extractive and abstractive. Figure 1 shows how each approach is implemented using various techniques. This section will give a fast description of each of these approaches, as well as the methods used in the literature for each approach.

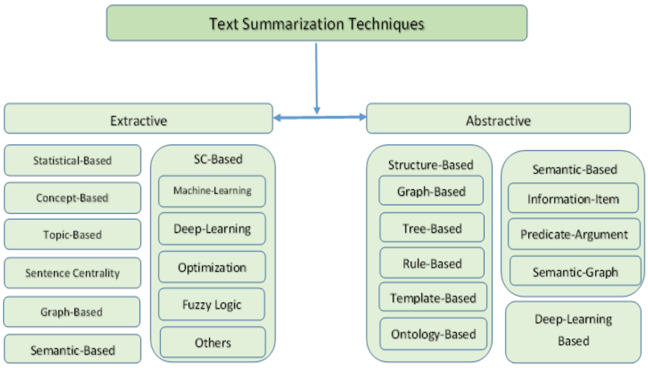

Figure 1. Automatic text summarization technique and their associated method

Abstractive summarization creates a broad summary by generating new phrases in the same way that a human being does. It is possible that the summary will include new phrases not included in the original text. Compression strategies and Language generation are required for generating abstractive summaries.
Abstractive text summarization in general categorized into three types: Structure based approach, semantic based approach and deep learning approach. The structure-based uses pre-existed structures for example, trees, graphs, rules, ontologies and templates. In addition, semantic-based approach uses natural language generation techniques and text semantic representation for example based on predicate arguments ,information items, and semantic graphs While deep learning approach supply another classification to the abstractive approach as classical or neural-based which in general indicates to any system that is not based on neural[25].
The extractive based summarizing method chooses informative sentences first from document as they appear in the source document based on specific criteria. Before extractive summarizing, the main challenge is determining which sentences from of the input document are significant and likely to be

included in summary. Sentence scoring based on sentence features used for this task [44]. It first, assigns a score to every sentence based on its features and then ranks the sentences based on their score. The highest-scoring sentences are most likely appear in the final summary. There are a variety of extractive text summarizing methods, such as statistical, concept-based, topic-based, clustering-based, graph-based, semantic-based, machine-learning based, deep learning based, fuzzy-logic-based and optimization-based.

Optimization-Based Methods

These methods formulated the summarization problem as an optimization problem. For example, in [26] a generic extractive multi-document ATS system is described as a multi- objective problem. There are two steps of sentence scoring. Firstly, a suitable representation for the input text is constructed. In this step, the vector representation is resulted (wherein every sentence in the entered text represented as a vector of words). Secondly, summary sentences is selected using an optimization algorithm such as the Multi-Objective Artificial Bee Colony (MOABC) algorithm Taking into account the length of the summary required and some criteria for optimization of the summary: coherence, reduced repetition, content coverage. The effectiveness of genetic algorithms in adjusting weights applied to ATS. The sentence scoring processes for genetic-algorithm for summarizer, according to [27], are as follows: 1) recognizing features of text in the entered text, such as sentences position, sentences length, and so on, 2) adjusting the weights of these features using the genetic algorithm, and then computing the sentence scores. There are many optimization algorithms such as GA, PSO, ACO, and ABC. Explained in the following paragraphs. Figure 2 show Flowcharts of these algorithms

Genetic algorithm is one of the well-known optimization algorithms. Its principle of operation depends on the chromosomes containing the genes, the length of the chromosome is initially determined after it encoded by a binary code. Parents select offspring probabilistically with a certain fitness function. The concept of crossover and mutation is specific to this algorithm. The genes of the parents exchanged with the crossover to generate new individuals. In mutation, the genes of the parents changed randomly in order to acquire new characteristics and not to fall into the local optima. Then the good ones replace the individuals with less fitness value. and then survivors are chosen [9].

In the work of Sim´on and others [7]. They proposed Genetic Algorithm (GA) for solving extractive summarization problem. Each individual represented as vector of positions of set of sentences. The fitness function based on both co-occurrence of bag-of-words and bigrams. For the selection process, the authors used elitism stage and tournament selection operator for this purpose. Which have the potential to select the better aptitude individuals and pass it to the future generations. Cycle crossover operator (CX) and insertion mutation operator used in this work. The used datasets are DUC01 and DUC02.

In the work of Castañeda and others [8], the authors developed a system for automatic extractive text summarization (EATS). The system in general is consisting of three parts. First part is feature generation they used a combination of methods. Which are TF-IDF, One-Hot Encoding (OHE), Doc2Vec, and Latent Dirichlet allocation (LDA). The second part is the proximity measures the authors used Euclidean distance and cosine similarity. The third part is cluster validation indexes, which is responsible for checking the quality of clustering. The best sentences summarizing the document set by using LDA.

In the work of Mojrian and Mirroshandel [9], they proposed the usage of Quantum Inspired Genetic Algorithm (QIGA) for this problem. In QIGA, the usage of superposition utilized. Meaning each individual can represent 2k states. Where k is the number of quantum bits (Q-bits). Resulting that the potential solutions in QIGA will be much larger than the population of classical GA with the same size. The fitness function is consisting of three main parts: First, combine the sentence scoring measures by using the term frequency, and sentence position. Second: combination of three sentence-scoring measures, which are inter-sentence cosine similarity of the input text, and between sentences and the title, and the sentence length measure. Third: which is the main objective function, is consist of six metrics. All of them are for scoring sentence. Four of them are using statistical methods. Two of them are using cosine similarity. The roulette wheel selection is used. Quantum bit-flip mutation is used. Elitism replacement used in this work.

The work of Chen and others [10], the authors first applied the tokenization step by using Moses Tokenizer. The words resulted from this step are converted into lower-case words. Each sentence then converted into array; each item in the array is representing the weight of a word in that sentence. The population size is set to 100 through all the iterations. This led to reducing the time cost. For the fitness function, an average of Rouge-1-n score is used. Two-point crossover used with 0.8 probability. Deletion mutation used, with probability of 0.01. Tournament selection used, with top five selection. The used dataset is the CNN / Daily Mail. After tuning the hyper-parameters of the experiments, the highest value of Rouge is 28.6.

In the work of V´azquez and others [11], they argued that successful extractive text summarization should pay deeper attention to all the steps of Pre-processing, term selection, term weighting, sentence weighting, and sentence selection. Instead of focusing on the last step only. The fitness function is multi-objective function, which based on sentence position, the sentence length, similarity with title, and position coverage feature. Roulette selection is used. Modified n-point crossover is used. For mutation, the inverse mutation operator selected. It applied twice. DUC01 and DUC02 datasets are used.

In the work of Gamal, El-Sawy, and AbuEl-Atta [12], they started by applying preprocessing steps. These steps are Sentence Segmentation, which is about separate each sentences apart. Tokenization, which is about separating each word apart. Stop words removal. Words Stemming. After that, Feature Extraction phase is considered. The authors of the work extracted the features: Title Feature, Sentence Length, Sentence Position, Numerical Data, and Thematic Words. Final score is the summation of the previous features. The Genetic Algorithm used to select the best combination of the sentences to give a good summary of the original text. The fitness function calculated based on the highest similarity between the sentences. The cosine relation between the different sentences are calculated. The quite criteria based on a threshold that is determined by the authors each time.

In the work of Al-Radaideh and Bataineh [13], the authors developed a system based on domain knowledge and Genetic Algorithm to do extractive summarization on Arabic language text. They started by the preprocessing steps. Get the keywords of the document (by using the domain knowledge). Final score for each sentence calculated based on Domain keywords, Term frequency, Sentence length, Sentence position, Title similarity, and Informative score. After that, a cosine similarity used for constructing the similarity between the sentences. Then the Genetic Algorithm used for detecting the best solutions. The fitness function based on the similarity of the sentences. Higher the similarity, higher the score. Tournament selection, one-point crossover and mutation rate (0.1). The used datasets are KALIMAT corpus, and Essex Arabic Summaries Corpus (EASC).

PSO is a continuous optimization technique inspired by swarming birds' collective behavior. Based on the associated velocity vector, all individuals (particles) that make up the population (swarm) fly throughout the search space. The most appealing of PSO feature is the retention of two crucial components: the global best (gbest) and the personal best (pbest).its Indicate to the positions in the swarm and for each particle. Here, gbest is an component that encourages to the convergence of swarm to the correct positions, while pbest responsible to the swarm's variety by creating unique behaviors for each particle. Both gbest and pbest are primarily utilize to update velocity while considering inertia. The function of velocity update is identical to the goal to-best. PSO differ from EA in that it does not include step of selection, with each iteration, the velocity and position updated only. This algorithm's simplistic structure makes coding and computations simple [15].

In the work of Mosa [14], the author aim is to summarize the social media comments. First, the comments obtained and converted into pivot words to keep only the important words in the original text. Second, the Graph Coloring theory used to generate several groups of the original text. With each group representing a possible solution. Third step, involved the use of Particle Swarm Optimization and Gravitational search algorithm together into hybrid algorithm to tackle multi-objective optimization problem. In the last step, the groups updated by using incremental algorithm. The used datasets are two datasets collected by the author from Facebook posts, comments, and messages of top 25 Arabic Facebook pages.

In the work of Priya and Umamaheswari [15], the steps for text summarization summarized as follow. First, the data preprocessing phase. Moreover, each document transformed into feature vector by using unigram BoW model. The second phase includes the feature extraction. In which Latent Semantic Analysis used, due to its ability to find the similarity relationship between the documents and the terms. Furthermore, TF-IDF is use for keywords extraction. The third phase includes sentiment extraction. In which Maximum Entropy Model used to calculate the strength of each feature. The phase four is the use of multi-objective Particle Swarm Optimization.

ACO is a meta-heuristic inspired by ant behavior that is primarily geared for combinatorial optimization issues. Artificial ants construct the solution cumulatively by adding the solutions not used in the solution space in the previous stage to the components of the current solution[28][29]. The decision to use the current solution or not is related to two main components, pheromone intensity and heuristic information (if available), which are considered the secret behind the success of the ant colony algorithm in solving optimization problems. Once the ants have selected a candidate solution, the pheromone density of this solution updated, this done by the two properties of evaporation and deposits. Artificial ants increase the pheromone intensity in their path based on the evaluation values. While the evaporation process continues for all the paths. The paths with a high amount of pheromone will attract ants and therefore this path will become the optimal solution; otherwise, the selection will soon become non- competitive. The pheromone update reflects the artificial ant colony's experience accumulated, this will increase the quality of the candidate solutions [1].Related works that is prominent and new in the proposed method or in characterizing, and the results will be discussed.

In the work of Al-Saleh and Menai [16], the authors used Ant Colony Optimization to generate summarization of English and Arabic texts. First phase is the pre-processing phase. Second, phase each word given a score. This done by using PageRank and HITS graph ranking algorithms. Three graphs built. First graph is bipartite graph links the word with the sentences. The edges given the value of TF-ISF and cosine similarity. The second graph is the relationship between each two of the sentences. The connection value calculated by using TF-ISF and cosine similarity too. Third graph calculate the relationship between each two words in the croups by using the longest common Substring. Third phase, the top score are used in Ant Colony Optimization to generate the summarization.

In the work of Lucky and Girsang [17], the authors used ACO to satisfy two objectives in the aim of text summarization. The two objectives are the lowest number of the sentences possible, and the highest accuracy of summarization as possible. The authors first collected data from Twitter by using Twitter API. Second, the text cleaned. Furthermore, sentence tokenization, word count, stop word removal, and stemming applied too. After that, the sentences converted into vectors by using bag1 of words model. Next, undirected graph is constructed. Where each sentence represented as the node of the graph. The edges of the graph constructed if only the value of cosine similarity between the two nodes are below threshold. Next, each edge given a value that calculated based on the cosine similarity, and the word count of the sentences. Furthermore, PageRank used too to rank each sentence. Next, ACO used to select the top sentences the most important ones.

In the work of Mosa, Hamouda, and Marei [18]. The authors used Ant Colony Optimization with Jensen-Shannon Divergence to summarize the text. The proposed work first converted the list of comments into list of term vector. Second, the construction of acyclic semi-graph performed. In this process, two conditions followed. First condition the very long text ignored. Second condition the higher the similarity between two sentences, the more desirable it is. Third, hybrid Ant Colony Optimization and Jensen-Shannon Divergence used to detect the more representative sentences.

ABC is a swarm intelligence-based stochastic search technique that simulates the behavior of honeybee swarms looking for food. Each candidate solution in this algorithm reflects the position of a source of food in the search space, with the quality of nectar amount employed as a fitness evaluator. It presents three group of bees: onlookers, employed bees, and scouts. The number of employed bees equals the source of food. Employed bees depart the hive in quest of a food source. Onlookers use the information provided by the employed bees to recruit a new food supply based on the selection probability of nectar quantity and leave the food source with low fitness value. When an onlooker bee chooses a new food source to discover, it becomes an employed bee. When the food source of employed bees abandoned available, it transforms into a scout bee, searching the search space at random for a food source. This procedure is iterated until the optimal source of food has been identified [3].

The work of Sanchez-Gomez, Vega-Rodríguez, and Pérez [19] used the ABC technique, based on decomposition for the purpose of text summarization. Which considered one of the multi-objective optimization methods. The two objectives used in this work are content coverage and redundancy reduction. They first pre-process the text. The algorithm is working by initializing the population first. This done by following three steps. A mutation operation with mutation ratio is set to 0.1 applied to generate new solution. The selection operation based on the decomposition method. After that, the onlooker bee will select the corresponding solution to it, by following the selection probability. The last step is to examine if the bees are not improved. The DUC2002 dataset is used. The evaluation metrics used in this work based on Rouge metrics. The average of the obtained results are 0.050 Range, and 8.87 CV.

In the work of Sanchez-Gomez, Vega-Rodrıguez, and Perez [20] carried out the problem of text summarization by representing the text as Vector-based word. The Cosine similarity considered as the first objective, and the second objective built from the factors of length, content coverage, and redundancy reduction. The input text is first pre-process by applying segmentation, tokenization, stop words removal, and stemming.

The later work of Sanchez-Gomez and his colleagues [21], they used Artificial Bee Colony again to solve the problem of text summarization. In this work, they followed the same steps as the previous one. However, they try of using parallelization techniques to run multiple experiments on the same time, and analysis the results. However, they followed the same configuration as the previous work. Although, this method is time consuming due to its repeating steps.

Table 1. Summary for result of the related work

Reference	Dataset	Result
[7]	DUC01	ROUGE-1:59.408
		ROUGE-2:33.422
	DUC02	ROUGE-1:62.367
		ROUGE2:35.742
[8]	DUC02	ROUGE-1:0.48681
		ROUGE-2:0.23334
		ROUGE-Su:0.24954
	TAC11	ROUGE-1:0.33682
	CNN/Daily mail	ROUGE-1:41.4
		ROUGE-2:18.4
		ROUGE-L:37.6
[9]	DUC05	ROUG-1:0.4106
		ROUGE-2:0.898
		ROUGE-SU4:0.1472
	DUC07	ROUGE-1:0.4767
		ROUGE-2:0.1287
		ROUGE-SU4:01885
[10]	CNN / Daily Mail	ROUGE:28.6
[11]	DUC01	ROUGE-1:0.4503
		ROUGE-2:0.1964
	DUC02	ROUGE-1:0.4842
		ROUGE-2:0.2247
[12]	CNN/Daily Mail	ROUGE-1:44.3
		ROUGE-2:21.17
[13]	Kalimat corpus	ROUGE-1: 0.528
		ROUGE-2: 0.407
	EAS corpus	ROUGE-1: 0.43
		ROUGE-2: 0.329
[14]	Facebook posts and comment	ROUGE-1: 0.951
		ROUGE-2: 0.934
		ROUGE-3: 0.929
[16]	MultiLing Pilot	ROUGE-1: 0.47397

eISSN 1303-5150

		ROUGE-2: 0.17737
		ROUGE-SU4: 0.21075
		ROUGE-L: 0.440136
[17]	data from Twitter	Cosine distance: 0.127
[18]	Facebook pages	Precession:0.947
		Recall: 0.889
		F-measure: 0.917
[19]	DUC02	ROUGE: 0.05
		CV: 8.87
[20]	DUC02	ROUGE: 0.027
		CV: 8.07
[21]	DUC	CV for ROUGE-2: 0.389
		CV for ROUGE-L: 0.581

IV. Evaluation methods

In the literature of Natural Language Processing, there are different evaluation metrics used for different operations and tasks. For the purpose of Text Summarization and Machine Translation, one of the most used and effective metrics is ROUGE. ROUGE stands for Recall-Oriented Understudy for evaluation. Which in general, is a set of methods used for comparing how good a text summarization or translation produced by machine with the text produced by humans.

There are few different types of Rouge metrics that each one follow slightly different approach than the other. In the following, each one will be discussed.

3.1. Rouge-N

The Rouge-N calculates the number of matching words or tokens between the machine generated and human generated (i.e. reference) text. The N in Rouge- N refers to 'n-grams'. If the number of n-grams is 1, it is referred to as unigram. In which each single word in the reference text is searched for in the generated text.

If the n is 2, this is referred to as bigram. In which every two words are matched together between the original and the generated text. Similarly, if the n is 3, it is referred to as trigram [30].

To measure the Rouge-n, if we use unigram (i.e. Rouge-1), we would calculate the matching rate between the generated text and the reference text.

3.2. Recall

The recall counts the overlapping number of n-grams in the generated text and the human-generated (reference) text, divided by the total number of n-grams. The equation shows how to calculate it.

$$recall = \frac{count_{match}(gram_n)}{count(gram_n)_{refrence}}$$

3.3. Precision

To overcome the problem of the recall metric, in precision similar equation is used. But instead of dividing by the total n-grams in the reference, it is divided by the total n-grams in the generated text. As shown in equation.

$$precision = \frac{count_{match}(gram_n)}{count(gram_n)_{generated\ \ text}} \qquad (2)$$

Figure 5 shows the same example used in the recall metric, but it is calculated by using the precision metric [74].

3.4. F1 score

The F1-score used to combine both the recall and precision metrics. Equation $ shows how to calculate it [74].

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall} \qquad (3)$$

The F1-score is calculating the coverage of the original text (by using recall), and making sure it is not covering irrelevant words (by using precision).

3.5. Rouge-L

In Rouge-L, the metric is uses the Longest Common

Subsequence (LCS) to measure the recall, precision, and F1-score instead of the n-grams. The LCS is the longest attached words that appears together in the generated and reference text [74].
The motivation behind the LCS is the longer the shared sequence is, the more similar the texts are.

3.6. Rouge-S

This known as “skip gram concurrence metric”. In which the recall, precision, and F1-score calculated by using predefined level of tolerance. For example, consider the examples in Figures 11and 12. In which the recall and precision calculated even where there was a word (i.e. brown) between the sentence in the reference and generated text. Since the Rouge-S can add a level of tolerance to skip some number of words if they are in the middle of the matched sentence [74].The Rouge-S is less popular than the previous discussed metrics.

3.7. Rouge discussion

Although Rouge is widely used in tasks such as Text Summarization and Machine Translation. However, there are few drawbacks of using it. For example if the generated text is similar to the reference text in meaning, but they are using synonyms words, the Rouge will fail to give a high similarity degree between the two texts.

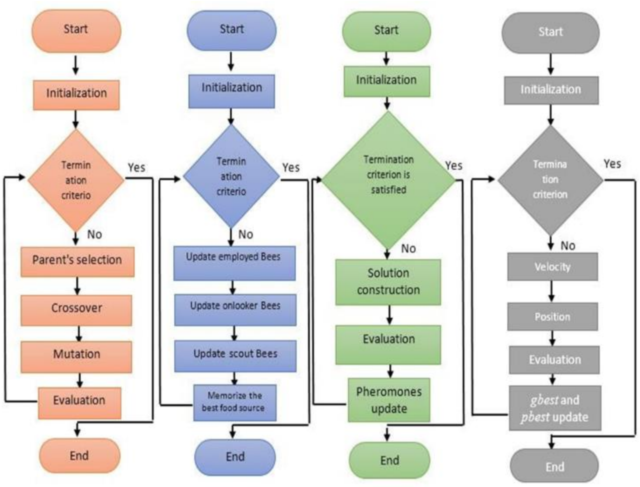

Figure 2. Flowcharts of GA, ABC, ACO, PSO

Conclusion

It is well-known that there is no one optimization model works best for all possible situations [31]. Based on that theorem, there are hundreds of optimization-based approaches to tackle the problem of text summarization reviewed in this paper. The effectiveness and shortcomings covered in the review of existing literature. Although it is not as near as the human ability to tackle this problem, optimization-based algorithms showed good results. However, there are further improvement to enhance the performance of such algorithms can be achieved if the research community pay more attention to the modeling of the text summarization problem.

References

A. Al-Saleh and M. E. B. Menai, “Solving multi-document summarization as an orienteering problem,” Algorithms, vol. 11, no. 7, 2018, doi: 10.3390/a11070096.

X.-S. Yang, “Metaheuristic Optimization,” Scholarpedia, vol. 6, no. 8, p. 11472, 2011, doi: 10.4249/scholarpedia.11472.

ABHISHEK SHARMA, ABHINAV SHARMA, SACHI CHOUDHARY, RUPENDRA KUMAR PACHAURI, AAYUSH SHRIVASTAVA, and DEEPAK KUMAR, “a Review on Artificial Bee Colony and Its Engineering Applications,” J. Crit. Rev. , vol. 7, no. 11, pp. 4097–4107, 2020, doi: http://www.jcreview.com/fulltext/197-1596854993.pdf?1597550239.

P. Verma and H. Om, “A novel approach for text summarization using optimal combination of sentence scoring methods,” Sadhana - Acad. Proc. Eng. Sci., vol. 44, no. 5, pp. 1–15, 2019, doi: 10.1007/s12046-019-1082-4.

T. Nakane, N. Bold, H. Sun, X. Lu, T. Akashi, and C. Zhang, “Application of evolutionary and swarm optimization in computer vision: a literature survey,” IPSJ Trans. Comput. Vis. Appl., vol. 12, no. 1, p. 3, Dec. 2020, doi: 10.1186/s41074-020-00065-9.

K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, “A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems,” IEEE/CAA J. Autom. Sin., vol. 6, no. 4, pp. 904–916, 2019, doi: 10.1109/JAS.2019.1911540.

J. R. Simón, Y. Ledeneva, and R. A. García-Hernández, “Calculating the significance of automatic extractive text summarization using a genetic algorithm,” J. Intell. Fuzzy Syst., vol. 35, no. 1, pp. 293–304, 2018, doi: 10.3233/JIFS-169588.

Á. Hernández-Castañeda, R. A. García-Hernández, Y. Ledeneva, and C. E. Millán-Hernández, “Language-independent extractive automatic text summarization based on automatic keyword extraction,” Comput. Speech Lang., vol. 71, no. May 2021, p. 101267, 2022, doi: 10.1016/j.csl.2021.101267.

M. Mojrian and S. A. Mirroshandel, “A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA,” Expert Syst. Appl., vol. 171, p. 114555, Jun. 2021, doi: 10.1016/j.eswa.2020.114555.

W. Chen, K. Ramos, and K. N. Mullaguri, “Genetic Algorithms For Extractive Summarization,” 2021, [Online]. Available: http://arxiv.org/abs/2105.02365.

E. Vázquez, R. Arnulfo García-Hernández, and Y. Ledeneva, “Sentence features relevance for extractive text summarization using genetic algorithms,” J. Intell. Fuzzy Syst., vol. 35, no. 1, pp. 353–365, 2018, doi: 10.3233/JIFS-169594.

M. Gamal, A. El-Sawy, and A. H. AbuEl-Atta, “Hybrid Algorithm Based on Chicken Swarm Optimization and Genetic

Algorithm for Text Summarization," Int. J. Intell. Eng. Syst., vol. 14, no. 3, pp. 319–331, 2021, doi: 10.22266/ijies2021.0630.27.

Q. A. Al-Radaideh and D. Q. Bataineh, "A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms," Cognit. Comput., vol. 10, no. 4, pp. 651–669, 2018, doi: 10.1007/s12559-018-9547-z.

M. A. Mosa, "A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining," Appl. Soft Comput., vol. 90, p. 106189, May 2020, doi: 10.1016/j.asoc.2020.106189.

V. Priya and K. Umamaheswari, "Enhanced continuous and discrete multi objective particle swarm optimization for text summarization," Cluster Comput., vol. 22, pp. 229–240, 2019, doi: 10.1007/s10586-018-2674-1.

A. Al-Saleh and M. E. B. Menai, "Ant Colony System for Multi-Document Summarization," in 27th International Conference on Computational Linguistics, 2018, pp. 734–744.

Lucky and A. S. Girsang, "Multi-objective ant colony optimization for automatic social media comments summarization," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 3, pp. 400–408, 2019, doi: 10.14569/IJACSA.2019.0100352.

M. A. Mosa, A. Hamouda, and M. Marei, "Ant colony heuristic for user-contributed comments summarization," Knowledge-Based Syst., vol. 118, pp. 105–114, Feb. 2017, doi: 10.1016/j.knosys.2016.11.009.

J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "A decomposition-based multi-objective optimization approach for extractive multi-document text summarization," Appl. Soft Comput. J., vol. 91, p. 106231, 2020, doi: 10.1016/j.asoc.2020.106231.

J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," Knowledge-Based Syst., vol. 159, pp. 1–8, 2018, doi: 10.1016/j.knosys.2017.11.029.

J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Parallelizing a multi-objective optimization approach for extractive multi-document text summarization," J. Parallel Distrib. Comput., vol. 134, no. June 2020, pp. 166–179, 2019, doi: 10.1016/j.jpdc.2019.09.001.

C. C. Yang and F. L. Wang, "Hierarchical summarization of large documents," J. Am. Soc. Inf. Sci. Technol., vol. 59, no. 6, pp. 887–902, Apr. 2008, doi: 10.1002/asi.20781.

S. Harabagiu and F. Lacatusu, "Using topic themes for multi-document summarization," ACM Trans. Inf. Syst., vol. 28, no. 3, pp. 1–47, Jun. 2010, doi: 10.1145/1777432.1777436.

L. Huang, Y. He, F. Wei, and W. Li, "Modeling Document Summarization as Multi-objective Optimization," in 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Apr. 2010, pp. 382–386, doi: 10.1109/IITSI.2010.80.

S. Gupta and S. K. Gupta, "Abstractive summarization: An overview of the state of the art," Expert Syst. Appl., vol. 121, pp. 49–65, May 2019, doi: 10.1016/j.eswa.2018.12.011.

J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Experimental analysis of multiple criteria for extractive multi-document text summarization," Expert Syst. Appl., vol. 140, p. 112904, Feb. 2020, doi: 10.1016/j.eswa.2019.112904.

Y. K. Meena and D. Gopalani, "Evolutionary Algorithms for Extractive Automatic Text Summarization," Procedia Comput. Sci., vol. 48, pp. 244–249, 2015, doi: 10.1016/j.procs.2015.04.177.

H. N. K. AL-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Hybrid Ant Colony Optimization and Genetic Algorithm for Rule Induction," J. Comput. Sci., vol. 16, no. 7, pp. 1019–1028, Jul. 2020, doi: 10.3844/jcssp.2020.1019.1028.

R. S. Hayder Naser Khraibet Al-Behadili, Ku Ruhana Ku-Mahamud, "ANT COLONY OPTIMIZATION ALGORITHM FOR RULEBASED CLASSIFICATION: ISSUES AND POTENTIAL SOLUTIONS," J. Theor. Appl. Inf. Technol., vol. 96, no. 21, pp. 7139–7150, 2018.

C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," Proc. Work. text Summ. branches out (WAS 2004), no. 1, pp. 25–26, 2004, [Online]. Available: papers2://publication/uuid/5DDA0BB8-E59F-44C1-88E6-2AD316DAEF85.

D. H. Wolpert, D. Nna, H. Road, S. Jose, and W. G. Macready, "No Free Lunch Theorems for Optimization 1 Introduction," pp. 1–32, 1996.