BiDETS: Binary Differential Evolutionary based Text Summarization

Hani Moetque Aljahdali[1]
Ahmed Hamza Osman Ahmed[2]
Department of Information System
Faculty of Computing and Information Technology
King Abdulaziz University Rabigh, Saudi Arabia

Albaraa Abuobieda[3]
Faculty of Computer Studies
International University of Africa, 2469
Khartoum, Sudan

***Abstract*—In extraction-based automatic text summarization (ATS) applications, feature scoring is the cornerstone of the summarization process since it is used for selecting the candidate summary sentences. Handling all features equally leads to generating disqualified summaries. Feature Weighting (FW) is an important approach used to weight the scores of the features based on their presence importance in the current context. Therefore, some of the ATS researchers have proposed evolutionary-based machine learning methods, such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), to extract superior weights to their assigned features. Then the extracted weights are used to tune the scored-features in order to generate a high qualified summary. In this paper, the Differential Evolution (DE) algorithm was proposed to act as a feature weighting machine learning method for extraction-based ATS problems. In addition to enabling the DE to represent and control the assigned features in binary dimension space, it was modulated into a binary coded format. Simple mathematical calculation features have been selected from various literature and employed in this study. The sentences in the documents are first clustered according to a multi-objective clustering concept. DE approach simultaneously optimizes two objective functions, which are compactness measuring and separating the sentence clusters based on these objectives. In order to automatically detect a number of sentence clusters contained in a document, representative sentences from various clusters are chosen with certain sentence scoring features to produce the summary. The method was tested and trained using DUC2002 dataset to learn the weight of each feature. To create comparative and competitive findings, the proposed DE method was compared with evolutionary methods: PSO and GA. The DE was also compared against the best and worst systems benchmark in DUC 2002. The performance of the BiDETS model is scored with 49% similar to human performance (52%) in ROUGE-1; 26% which is over the human performance (23%) using ROUGE-2; and lastly 45% similar to human performance (48%) using ROUGE-L. These results showed that the proposed method outperformed all other methods in terms of F-measure using the ROUGE evaluation tool.**

Keywords*—*Differential evolution; text summarization; PSO; GA; evolutionary algorithms; optimization techniques; feature weighting; ROUGE; DUC

I. Introduction

Internet Web services (e.g. news, user reviews, social networks, websites, blogs, etc.) are enormous sources of textual data. In addition, the collections of articles of news, novels, books, legal documents, biomedical records and research papers are rich in textual content. The textual content of the web and other repositories is increasing exponentially every day. As a result, consumers waste a lot of time seeking the information they want. You cannot even read and understand all the search results of the textual material. Many of the resulting texts are repetitive or unimportant. It is also necessary and much more useful to summarize and condense the text resources. Handbook summary is a costly and time- and effort-intensive process. In fact, manually summarizing this immense volume of textual data is difficult for humans [1]. The main solution to this problem is the Automated Overview Text (ATS).

ATS is one of the information retrieval applications which aim to reduce the amount of text into a condensed informative form. Text summarization applications are designed using several and diverse approaches, such as “feature scoring”, “cluster based”, “graph based” and other approaches. In the “feature scoring” based approach, some research proposals can be divided into two directions. The first direction concerns proposing features of either novel single feature or structured features. Researchers who are working on this first direction claim that existing features are poorer and not eligible to produce a qualified summary. The second direction is concerned with proposing mechanisms aiming to adjust the scores of already existing features by discovering their real weight (importance) in the texts. This direction claims that employing feature selection may be considered a good solution rather than introducing novel features. Many researchers have proposed feature selection methods, while a limited number of them utilized the optimization systems with the purpose of enhancing the way the features were commonly used to be selected and weighted.

The extractive method extracts and uses the most appropriate phrases from the input text to produce the description. The Abstractive approach represents an intermediate type for the input text and produces a description of words and phrases which differ from the original text phrases. The hybrid approach incorporates extraction and abstraction. The general structure of an ATS system comprises:

1) Pre-processing: development by using several linguistic techniques, including phrase segmentation, word tokenization, stop word deletion, speech marking, stemming and so forth of a standardized representation of the original text [2].

2) Process: use one or more summary text methods to transform the input document(s) into the summary by applying one technique or more. Section 3 describes the various ATS methods and Section 4 discusses the various strategies and components for implementing an ATS framework.

3) Post-processing: solve some problems in summary sentences generated, such as anaphora resolution, before generating a final summary and repositioning selected sentences.

Generating a high quality text summary implies designing methods attached with powerful feature-scoring (weighting) mechanisms. To produce a summary of the input documents, the features are scored for each sentence. Consequently, the quality of the generated summary is sensitive to those nominated features. Consequently, evolving a mechanism to calculate the feature weights is needed. The weight method aids in identifying the significance of features distinctly in the collection of documents and how to deal with them. Many scholars have suggested feature selection methods based on optimization mechanisms such as GA [3] and PSO [4]. This paper follows the same trend of these research studies and employed the unselected evolutionary algorithm "Differential Evolution" (DE) [5] to act as feature selection scoring mechanism for text summarization problems. For more significant evaluation, the authors have benchmarked the results of those evolutionary algorithms (PSO and GA) found in the related literature.

In this research, the DE algorithm has been proposed to act as a feature weighting machine learning method for extraction-based ATS problems. The main contribution of the proposed method is adopt the DE to represent and control the assigned features in binary dimension space, it was modulated into a binary coded format. Simple mathematical calculation features have been selected from various literature and employed in this study. The sentences in the documents are first clustered according to a multi-objective clustering concept. DE approach simultaneously optimizes two objective functions, which are compactness measuring and separating the sentence clusters based on these objectives. In order to automatically detect a number of sentence clusters contained in a document, representative sentences from various clusters are chosen with certain sentence scoring features to produce the summary. In general, the validity index of the clusters tests in various ways certain inherent cluster characteristics such as separation and compactness. Any sentences from each cluster are extracted using multiple sentence labeling features to generate the summary after producing high quality sentence clusters.

In this study, five textual features have been selected according to their effective results and simple calculations as stated in Section 4.1. To enable the DE algorithm to achieve the optimal weighting of the selected features, the chromosome (a candidate solution) was modulated into a binary-code format. Each gene position represents a feature. If the gene holds the binary '1' this means the equivalent feature is active and should be included in the scoring process, otherwise '0' means the corresponding features are inactive and should be excluded from the scoring process. Based on active and inactive features, each chromosome is now able to generate and extract a summary. A set of 100 documents was imported from the DUC 2002 [6] and [7] . The summary will be evaluated using the ROUGE toolkit [8] and the recall value would be assigned as a chromosome fitness value. After several iterations the DE extracts weights for each feature and takes them back again to tune the feature scores. Then the new and optimized summary is generated. Section 4 presents deep details on algorithm set-up and configurations.

Referring to the evolutionary algorithm's competition events, the DE algorithm showed powerful performance in terms of discovering the fittest solution in 34 broadly used benchmark problems [9]. This paper stressed the emphasis to use the unselected "DE" algorithm for performing FS process for ATS applications and established comparative and competitive findings with previously mentioned optimization based methods PSO and GA. The objective of the experimentation that was implemented in this study is to examine the capability of the DE when performing the feature selection process compared to other evolutionary algorithms (PSO and GA) and other benchmark methods in terms of qualified summary generation. The authors used a powerful DE experience to obtain high quality summaries and outperform other parallel evolutionary algorithms. Improving the performance significantly depends on providing optimal solutions for each generation of DE procedures. To do so as recent genetic operators we have taken into account existing developments in Feature Weighting (FW). The polynomial mutations concept is also used to enhance the discovery of the method suggested. The principal drawback in the text summarization, mainly in the short document problem, is redundant. Some researchers exploited the issue of redundancy by selected the sentences at the beginning of the paragraph first and calculated the resemblance to the following sentences to nominate the best. The Maximal Marginal Significance method is then recommended in order to minimize redundancies in multi-documentary and short text summarization, in order to achieve optimum results.

The rest of this study is presented as follows. Section 2 presents the literature review. An overview to the DE optimization method is introduced in Section 3. The methodology is detailed in Section 4. The proposed Binary Differential Evolution based Text Summarization (BiDETS) model is explained in Section 5. Section 6 concludes the study.

II. Related Work

Much research for the functions selection (FS) method has recently been proposed. Because of its relevance, FS affects application quality [10]. FS attempts to classify which characteristics are relevant and which data can reflect. In [11] the authors demonstrated that the device can minimize the problem's dimensionality, eliminate unnecessary data and uninstall redundant features by embedding FS. FS also decreases the quantity of data required and increases thereby the quality of the system results through the machine learning process.

A number of articles on ATS systems and methodologies have been published recently. Most of these studies concentrate on extractive summary techniques and methods, such as Nazari and Mahdavi [12], since the abstract summary requires a broad NLP. Kirmani et al. [13] describe normal statistical features and extractive methods. The surveys of the ATS extractive systems that apply fizzy logic methods in Kumar and Sharma [14] are given. Mosa et al. [15] surveyed how swarm intelligence optimization methods are used for ATS [15]. They aim to motivate researchers, particularly for short text summaries, to use ATS swarm intelligence optimisation. A survey on extractive deep-learning text summarization is provided by Suleiman and Awajan [16]. Saini et al.[17] Introduced a method attempting to develop several extractive single document text summarization (ESDS) structures with MOO frameworks. The first is a combination of the SOM and DE (called the ESDS SMODE) second is a multi-objective wolf optimizer (ESDS MGWO) based on multi-objective water mechanism and third a multi-objective water cycle algorithm (ESDS MWCA) based on a threefold approach. The sentences in the text are first categorized using the multi-objective clustering concept. The MOO frame simultaneously optimizes two priorities functions calculating compactness and isolation of sentence clusters in two ways. In some surveys, the emphasis is upon abstract synthesis including Gupta and Gupta [18], Lin and Ng [19] for various abstract methods and the abstract neural network methodology methods [20]. Some surveys concentrate on the domain-specific overview of the documents such as Bhattacharya et al. [21] and Kanapala, Pal and Pamula [22], and abstractive deep learning methodologies and challenges of meeting summarization to confront extractive algorithms used in the microblog summarization [23]. Some studies presented and discussed the analysis of some abstractive and extractive approaches. These studies included details on abstract and extractive on resume assessment methods [24, 25].

Big data in social media was re-formulated for the extractive text summarization in order to establish a multi-objective optimization (MOO) mission. Recently, Mosa [26] proposed a text summarization method based on Gravitational Search Algorithm (GSA) to refine multiple expressive targets for a succinct SM description. The latest GSA mixed particle swarm optimization (PSO) to reinforce local search capacities and slow GSA standard convergence level. The research is introduced as a solution of capturing the similarity between the original text and extracted summary Mosa et al. [15, 27, 28]. To solve this dilemma in the first place, a variety of dissimilar classes of comments are based on the coloring (GC) principle. GC is opposed to the clustering process, while the separation of the GC module does not have a similarity dependent on divergence. Later on, the most relevant remarks will be picked up by many virtual goals. (1) Minimize an inconsistency in the JSD method-based description text. (2) Boost feedback and writers' visibility. Comment rankings depend on their prominence, where a common comment close to many other remarks and delivered by well-known writers gives emotions. (3) Optimize (i.e. repeat) the popularity of terms. (4) Redundancy minimization (i.e. resemblance). (5) Minimizing the overview planning. The ATS Single-Focus Overview Method for encoded extractor network architecture is proposed by Chen and Nguyen [29] using an ATS reinforcement learning algorithm and the RNN sequence model. A selective encoding technique at the sentence level selects the relevant features and then extracts the description phrases. S. N and Karwa. Chatterje[30] recommended an updated version and optimization criterion for extractive text summarization based on the Differential Evolution (DE) algorithm. The Cosine Similarity has been utilized to cluster related sentences based on a suggested criterion function intended to resolve the text summary problem and to produce a summary of the document using important sentences in each cluster. In the Discrete DE method, the results of the tests scored a 95.5% increase of the traditional DE approach over time, whereas the accuracy and recalls of derived summaries were in all cases comparable.

In text processing applications, several evolutionary algorithms based “feature selection” approaches were widely proposed, in particular text summarization applications. To select the optimal subset of characteristics, Tu et al. [31] used PSOs. These characteristics are used for classification and neural network training. Researchers in [32] have selected essential features using PSO for text features of online web pages. In order to strengthen the link between automated assessment and manual evaluation, Rojas-Simón et al. [33] proposed a linear optimization of contents-based metrics via genetic algorithm. The suggested approach incorporates 31 material measurements based on the human-free assessment. The findings of the linear optimization display correlation gains with other DUC01 and DUC02 evaluation metrics. In 2006, Kiani and Akbarzadeh [34] presented an extractive-based automatic text summarization system based on integration between fuzzy, GP and GA. A set of non-structural features (six features) are selected and used. The reason behind using the GA is to optimize the membership function of fuzzy, whereas the GP is to improve the fuzzy rule set. The fuzzy rule sets were optimized to accelerate the decision-making of the online Web Summarizer. Again, running several rule sets online may result in high time complexity. The dataset used to train the system was three news articles which differed in their topics, while any article among them could be used for the testing phase. The fitness function is almost a total score of all features combined together. Ferreira, R et al. [35] Implemented a quantitative and qualitative evaluation research to describe 15 text summarization algorithms for sentence scoring published in the literature. Three distinct datasets such as Blogs, Article contexts, and News were analysed and investigated. The study suggested a new ways to enhance the sentence extraction in order to generate an optimal text summarization results.

Meanwhile, BinWahlan et al. [36] presented the PSO technique to emphasize the influence of structure of feature on the feature selection procedure in the domain of ATS. The number of selected features can be put into two categories, which are “complex” and “simple” features. The complex features are “sentence centrality”, “title feature”, and “word sentence score”; while the simple features are “keyword” and “first sentence similarity”. When calculating the score of each feature, the PSO is utilized to categorize which type of feature is more effective than another. The PSO had encoded typically to the number of used features; and the PSO modulated into binary format using the sigmoid function. The score of ROUGE-1 recall was used to compute the fitness function value. The dataset used consisted of 100 DUC 2002 papers for machine preparation. Initialization of the PSO parameters and measurement of best values derived weight from each characteristic. The findings showed that complex characteristics are weighted more than the simple ones, suggesting that the characteristic structure is an important part of the selection process. To test the proposed model the authors published continued works as found in [36]. In addition, the dataset was split into training and testing phases in order to quantify the weights[36]. Ninety-nine documents have been assigned to shape the PSO algorithm while the 100th was assigned to evaluate the model. Therefore, the phrases scored are rated descending, where n is equivalent to a summary duration; the top n phrases are selected as a summary. In order to test the effects, the authors set a human model description and the second as a reference point. For comparison purposes a Microsoft Word-Summary and a first human summary were considered. The results showed that PSO surpassed the MS-Word description and reached the closest correlation with humans as did MS-Word. The authors [37] presented a genetic extractive-based multi-document summarization. The term frequency featured in this proposed work is computed not only based on frequency, but also based on word sense disambiguation. A number of summaries are generated as a solution for each chromosome, and the summary with the best scores of criteria is considered. These criteria include "satisfied length", "high coverage of topic", "high informativeness", and "low redundancy." The DUC 2002 and 2003 are used to train the model, while the DUC 2004 is used for testing it. The proposed GA model had been compared against systems that participated in DUC2004 competition-Task2. It achieved good results and outperformed some proposed methods. Zamuda, Aleš, and Elena Lloret[38] Proposed text summarization method to examine a machine linguistic problem of hard optimization based on multi-documents by using grid computing. Multi-document summarization's key task is to successfully and efficiently derive the most significant and unique information from a collection of topic-related, limited to a given period. During a Differential Evolution (DE), a data-driven resuming model is proposed and optimized. Different DE runs are spread in parallel as optimization tasks to a network in order to achieve high processing efficiency considering the challenging complexity of the linguistic system.

Two text summarization methods: adapted corpus base method (MCBA) and latent semantic analysis based text relationship map (TRM based LSA) have been addressed by [39]. Five features were used and optimized using GA. The GA in this study is so not highly different to the previous work, in which the GA was employed for feature weights extraction. The F-measure was assigned as a fitness value and the chromosome number of each population was set to 1000. The top ten fittest chromosomes shall be selected in the next round. This research introduced a lot of experiments using different compression rate effectiveness. The GA provided an effective way of obtaining features weights which further led the proposed method to outperform the baseline methods.

The GA was also used in the work of Fattah [40]. The GA was encoded and described as similar to [20], but the feature weights were conducted to train the "feed forward neural network" (FFNN), the "probabilistic neural network" (PNN), and "Gaussian mixture model" (GMM). The models were trained by one language, and the summarization performance had been tested using different languages. The dataset is calm of 100 Arabic political articles for training purposes, while 100 English religious articles were used for testing. The fitness value was defined as an average precision, and the fittest 10 chromosomes were selected from the 1000 chromosomes. In order to obtain a steady generation, the researchers adapted the generation number to 100 generations. The GA is then tested using a second dataset. Based on the obtained weights which were used to optimize the feature scores, the sentences were ranked in order to be selected for summary representation. The model was compared against the work presented by [39]. The results showed that the GA performed well compared to the baseline method. Some researchers have successfully modulated the DE algorithm from real-coded space into binary-coded space in different applications. G. Pampara et al. [41] introduced Angle-Modulation DE (AMDE) enabling the DE to work well in binary search space. Pampara et al. were encouraged to use the AM as it abstracts the problem representation simply and then turns it back into its original space. The AM also reduces the problem dimensionality space to "4-dimension" rather than the original "n-dimensional" space. The AMDE method outperformed both AMPSO and BinPSO in terms of required number of iterations, capability of working in binary space and accuracy. He et al. [10] applied a binary DE (BDE) to extract a subset feature using feature selection. The DE was used to help find a high quality knowledge discovery by removing redundant and irrelevant data, and consequently improving the data mining processes (dimension reduction). To measure the fitness of each feature subset, the prediction of a class label along with the level of inter-correlation between features was considered. The "Entropy" or information gain is computed between features in order to estimate the correlation between them. The next generation is required to have a vector with the lowest fitness function. Khushaba et al. [11] implemented DE for feature selection in Brain-Computer Interface (BCI) problem. This method showed powerful performance in addition to low computational cost compared to other optimization techniques (PSO & GA). The authors of this current study observed that the chromosome encoding is in real format but not in binary format. A challenge facing Khushaba et al. was the appearance of "doubled values" while generating populations. To solve this problem, they proposed to use the Roulette-Wheel concept to skip such occurrences of double values.

In general, and in terms of evolutionary algorithm design requirement competition, the researchers are required to design algorithms which provide ease of use, are simple, robust and in line generate optimized or qualified solution [5]. It was found that the DE, due to its robust and simple design, outperformed other evolutionary methods in terms of reaching the optimal and qualified solution [9]. Also, the literature showed that text summarization feature selection methods which were based on evolutionary algorithms were limited to techniques of Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA). From this point of view, the researchers of this study had been encouraged to employ the DE algorithm for the text summarization feature selection problem.

Although there was successful implementation and high quality results obtained by DE, the literature showed that the DE had never been presented to act as Automatic Text Summarization feature selection mechanism. Section 4 discusses the experimental set-up and implementation in more detail. It is worth mentioning that the DE was presented before for a text summarization problem to handle the process of sentence clustering instead of feature selection [42, 43]. The DE was widely presented to handle object clustering problems such as [44-46] which is not the concern in this study.

III. Differential Volution Method

DE was originally presented by Storn and Price [5]. It is considered a direct search method that is concerned with minimizing the so-called "cost/objective function". Algorithms in heuristic search method are required to have a "strategy" in order to generate changes in parameter vector values. The Differential Evolution performance is sensitive to the choice of mutation strategy and the control parameters [47]. For the newly generated changes, these methods use the "greedy criterion" to form the new population's vectors. A decision of governing is simple, if the newly generated parameter vector participates in decreasing the value of the cost function then it will be selected. For this use of greedy criterion, the method converges faster. For the methods concerned with minimization, such as GA and DE, there are four features they should come over: ability of processing a non-linear objective function, direct search method (stochastic), ability of performing a parallel computation, and fewer control variables.

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

The DE utilizes a dimensional vector $x_{i,G}$ with size NP as shown in Formula 1, such that:

$$\mathrm{x}_{i,G} = 1,\ 2,\ \ldots,\ \mathrm{NP} \tag{1}$$

Random selections of the initial vector are rendered by summarizing the values of the random deviations from the nominal solutions Xnom,0. For producing new parameter vector, DE randomly finds the differences between two population vectors summing them up with a third one to generate what is the so-called Mutant vector. Equation 2 shows how to mutate vector Vj,G+1 from target vectors Xj,G.

$$\mathrm{v}_{i,\, G+1} = \mathrm{x}_{r_1,G} + \mathrm{F}.\left(\mathrm{x}_{r_2,G} - \mathrm{x}_{r_3,G}\right) \tag{2}$$

Where r1, r2, and r3 ∈ [1, 2… NP] are random index selections of type integer and, $r_1 \neq r_2 \neq r_3$, mutually different and F>0, for i≥4. F is const and real factor ∈ [0, 2]. The F is used to govern the augmentation of differential variation as in (3).

$$F.\left(\mathrm{x}_{r_2,G} - \mathrm{x}_{r_3,G}\right) \tag{3}$$

The trail or crossed vector is a new vector merged with a predefined parameter vector "target vector" in order to generate a "trail vector". The goal of crossover is to find some diversity in the perturbed parameter vectors. Equation (4) shows trail vector.

$$\left(v_{1i,G+1},\ v_{2i,G+1},\ \ldots,\ v_{Di,G+1}\right) \tag{4}$$

such that

$$\begin{cases} \mathrm{v}_{ji,G+1}\ if\ \left(\mathrm{randb}\left(\mathrm{j}\right) \leq \mathrm{CR}\right) or\ j = rnbr\left(i\right) \\ \mathrm{x}_{ji,G+1}\ if\ \left(\mathrm{randb}\left(\mathrm{j}\right) > \mathrm{CR}\right) or\ j \neq rnbr\left(i\right) \end{cases} \tag{5}$$

Where randb(j) is the jth evaluation of a uniform random number producer with the outcome ∈[0,1], CR, is the constant crossover ∈[0,1] and can be defined by the user, and rnbr(i) is a randomly selected value $\mathrm{i} \in 1, 2, \ldots, \mathrm{D}$ computed to guarantee that trail vector Vj,G+1 will obtain as a minimum one parameter from the mutated vector Vj,G+1.

In the Selection phase, if the trail vector obtained a lower value of cost function compared with the target vector, it will be replaced with the target vector in the next generation. In this step, the greedy criterion test takes place. If the trail vector Vj,G+1 yields a lower cost function than a target vector Vj,G+1 is assigned by selecting the trailed vector, or else nothing is done.

IV. Methodology

This section covers the proposed methodology (experimental set-up). Firstly, Section 4.1 presents the selected features needed to score and select the "in-summary" sentences and exclude the "out-summary" sentences. The subsequent three Sections 4.2 to 4.4 are about configuring the DE algorithm and show how the evolutionary chromosome has been configured and encoded; how the DE's control parameters were assigned; the suitable assignment of objective (fitness) function when dealing with text summarization problem. The selected dataset and principle pre-processing steps were introduced in Section 4.5. Section 4.6 discusses the sets of the selected benchmarks and other parallel methods for more significance comparison. The last section presents the evaluation measure that was used.

A. The Selected Features

A variety of text summarization features were suggested in order to nominate outstanding text sentences. In the experimental phase, five statistical features were selected to test the model [40, 48]:, Sentence-Length (SL) [49], Thematic-Word (TW) [50-52], Title-Feature (TF) [51], Numerical-Data (ND) [40], and Sentence-Position (SP) [51, 53].

1) Sentence Length (SL): The topic of the article may not be a short sentence. Similarly, the selection of a very long sentence is not an ideal choice because it requires non-informational words. A division by the longest sentence solves this problem to avoid choosing sentences either too short or too long. This normalization informs us equation (6).

$$\mathrm{SL}\left(\mathrm{S_i}\right)=\frac{\#\text{ of words in }\mathrm{S_i}}{\#\text{ of words in longest sentence}} \tag{6}$$

Where $\boldsymbol{S_i}$ refers the i^{th} sentence in the document.

2) Thematic Words (TW): The top n terms with the highest frequencies are a list of the top n terms chosen. In the first place, count frequencies in the documents measure the thematic terms. Then a threshold is set for the signing of which terms as thematic words should be chosen. In this case, as shown in equation (7), the top ten frequency terms will be chosen.

$$\mathrm{TW}-\mathrm{S_i}=\frac{\#\text{ of thematic words in }\mathrm{S_i}}{\text{Max number of TW found in a sentence}} \tag{7}$$

Where S_i refers to the $\boldsymbol{i^{th}}$ sentence in the document.

3) Title Feature (TF): To generate a summary of news articles, a sentence including each of the “Title” words is considered an important sentence. Title feature is a percentage of how much the word of a currently selected sentence i match words of titles. Title feature can be calculated using Equation (8).

$$\mathrm{TF}\left(\mathrm{S_i}\right)=\frac{\#\text{ of}\left(\mathrm{S_i}\right)\text{ words matched title words}}{\#\text{ of Title's words}} \tag{8}$$

Where Si refers to the *ith* sentence in the document.

4) Numerical Data (ND): A term containing numerical information refers to essential information, such as event date, money transaction, percentage of loss etc. Equation (9) illustrates how this function is measured.

$$\mathrm{ND}\left(\mathrm{S_i}\right)=\frac{\#\text{ of numerical data in }\mathrm{S_i}}{\text{Sentence Length}} \tag{9}$$

Where Si refers to the ith sentence in the document, and sentence length is the total number of words in each sentence.

5) Sentence Position (SP): In the first sentence, a significant sentence and a successful candidate for inclusion in the summary is considered. The following algorithm is used to measure the SP feature as shown in Equation (10).

For i = 0..t do

$$\mathrm{SP}\left(\mathrm{S_i}\right)=\frac{\mathrm{t-\ i}}{\mathrm{t}} \tag{10}$$

Where $\boldsymbol{S_i}$ refers to the $\boldsymbol{i^{th}}$ document sentence, and *t* is the total number of sentences in document *i*.

B. Configuring up DE: Chromosome, Control Parameters, and Objective Function

Mainly, this study focuses on finding optimal feature weights of text summarization problems. The chromosome dimension was configured to represent these five features. At the start, each gene is initialized with a real-coded value. To perform feature selection process, the need for modulating these real-codes in binary-codes was emerged. This study follows the same modulation adjustment presented by He et al. [10] as shown in Formula 11.

$$\mathrm{G_{x,y}}=\begin{cases}1,\ \mathit{rand}\left(\mathrm{r}\right)\geq \mathit{exp}\left(\left|\mathrm{x}\right|\right)\\ \quad 0,\ \mathit{otherwise}\end{cases} \tag{11}$$

Where $G_{x,y}$ refers to the current binary status of gene y in chromosome x, $rand()$ is a random function that generates a number $r \in (1,0)$, and $exp(|x|)$ is the exponential value of current gene x. If $rand(r)$ is greater than or equal to $exp(|x|)$ then $G_{x,y}=1\ else=1$ for each x in y.

If x=1 is modulated, it's activated and counted to the final score, or if the bit has zero then it is inactive and is not considered at the final score. The corresponding trait would not be considered. The chromosome structure of the features is shown in Fig. 1. The first bit denotes the first feature “TF”, the second bit denotes the second feature “SL”, the third bit denotes to the third feature “SP”, the fourth bit denotes the fourth feature “ND”, and the fifth bit denotes the fifth feature “TW”.

A chromosome is a series of genes; their value is status controlled through binary probability appearance. So all probable solutions will not exceed the limit of 2n where 2 refers to the binary status[0,1] and n is the problem dimension. Within this limited search space, the DE is suggested to cover all these expected solutions. In addition, it enables DE to assign a correct fitness to a current chromosome. For more explanation check a depiction example of a chromosome shown in Fig. 2. The DE runs real-coded mode ranges between (0,1). To assign a fitness to this chromosome in its current format may become a difficult task; check “row 1” at Fig. 2. From this point of view, a “modulation layer” is needed to generate a corresponding chromosome. Values of Row 1 are [0.65, 0.85, 0.99, 0.21, 0.54], then they will be modulated into binary string as shown at “Row 2” [0, 1, 1, 0, 1]; this modulation tells the system to generate a summary only based on the active features [F2, F3, and F4] and ignores the inactive features [F1 and F4]. Then the binary string itself (01101) is modulated into the decimal numbering system, which is equal to (13). Inline to this, the system will store the correspondent summary recall value in an indexed fitness file at position (13). Now, DE is correctly able to assign fitness value to a current binary chromosome of [0, 1, 1, 0, 1].

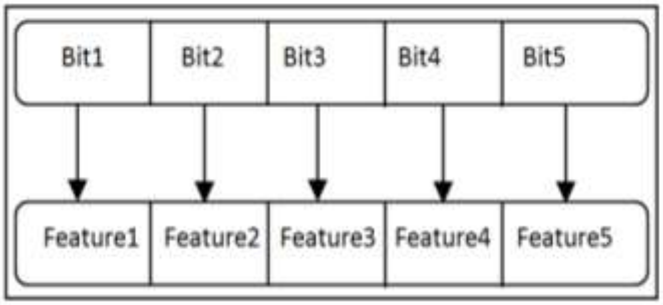

Fig. 1. Chromosome Structure.

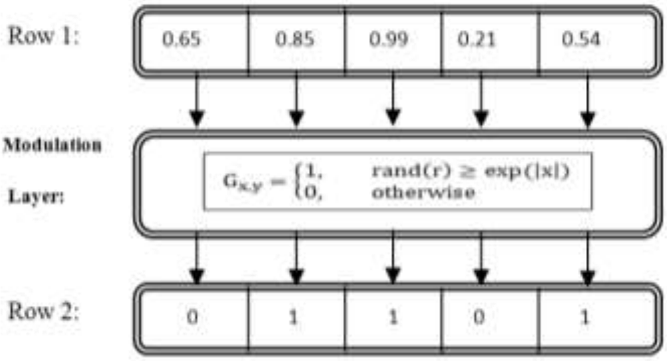
Row 1:
0.65
0.85
0.99
0.21
0.54
Modulation Layer:
$G_{x,y} = \begin{cases} 1, & \text{rand}(r) \geq \exp(|x|) \\ 0, & \text{otherwise} \end{cases}$
Row 2:
0
1
1
0
1

Fig. 2. Gene's Process.

The DE's control parameters were set according to optimal assignments found in the literature as follows. The F-value was set to 0.9 [5, 9, 54-57], the CR was set to 0.5 [5, 9, 54-57] and the size of population NP was set to 100 [5, 9, 54, 57-59]. It is widely known that optimization techniques are likely to run broadly within a high number of iterations such as 100, 500, 1000 and so on. In this experiment it was noted that DE is able to reach the optimal solution within a minimum number of iterations (=100) if the dimension length is so small. Empirically this study justified that when the dimension of the problem is small the number of total candidate solutions cannot be too large. The DE was tested and outperformed many optimization techniques in challenges of high dimension, fast convergence and extraction of qualified solution. To this end, the conducted experiment of this study approved that DE can reach the optimal solution within a very small number of iterations.

The fitness function is a measurement unit for techniques of optimization. These techniques are used to determine the chromosomes achieving the best and best solution. The new population of this chromosome can be restored (survived) in the next generation. This study generates only a probability of 2n chromosomes for each input document where n is the dimension or number of features. The system then assigns a fitness value for each chromosome of the resumes it generates. The highest recall value of the top chromosome is selected and the corresponding document will be shown in the dataset. In the literature, a similar and successful work [60] assigned the recall of ROUGE-1 as a fitness value. Equation 12 demonstrates how to measure the recall value in comparison to the reference summary for each summary generated.

C. Selected Dataset and Pre-Processing Steps

The National Institute of Standards and Technology of the U.S. (NIST) created the DUC 2002 evaluation data which consists of 60 data sets. The data was formed from the TREC disks used in the Question-Answering in task TREC-9. The TREC is a retrieval task created by NIST which cares about Question-Answering systems. The DUC 2002 data set came with two tasks single-document extract\abstract and multi-document extract\abstract with different topics such as natural disaster and health issues. The reason for using DUC 2002 in this study is that it was the latest data set produced by NIST for single-document summarization.

The selected dataset is composed of 100 documents collected from the Document Understanding Conference (DUC2002) [7]. These 100 documents were allocated on 10 clusters of certain topics labelled with: D075b, D077b, D078b, D082a, D087d, D089d, D090d, D092c, D095c, and D096c. Every cluster includes 10 relevant documents comprising 100 documents. Each DUC2002 article was attached with two "model" summaries. These two model summaries were written by two human experts (H1 and H2) and were of the size of 100 words for single document summarization.

Often text processing application datasets are exposed to pre-processing steps. These pre-processing steps include, but are not limited to: removal of stop words within the text, sentence segmentation, and stemming process based on porter stemmer algorithm [61]. One of the main challenges in text engineering research is segmenting sentences by discovering correct and unambiguous boundaries. In this study, the authors manually segmented the sentences to skip falling into any hidden segmentation mistakes and guarantee correct results. According to the selected methodology of the specific application sign or resign implementing the pre-processing steps is an unrestricted option. For example, some research of semantic text engineering applications may tend and prefer to retain the stop words and all words in their current forms not in their root forms (stemming). In this study, the mentioned pre-processing steps were employed.

D. The Collection of Compared Methods

This paper diversifies the selection of comparative methods to create a competitive environment. The compared methods had been divided into two sets. Set A includes similar published optimization based text summarization methods: Particle Swarm Optimization (PSO) [60] and Genetic Algorithm (GA) [62]. This set was brought in to add a significant comparison as this study also proposes the Differential Evolution (DE) for handling the FS problem of the text summarization. To the best of the author's knowledge the DE have never been presented before to tackle the text summarization feature selection problem. In the literature, the DE has been proposed before as sentences clustering approach of text summarization problem [42], but not for the feature selection problem. Both works GA and PSO are presented for the problem of feature selection in text summarization. Set B consists of DUC 2002 best system [63] and worst system [64]. Due to source code unavailability of methods in sets A and B, the average evaluation measurement results were being considered as published. In addition, and for a fair comparison, the BiDETS system had been trained and tested using the same dataset source (DUC2002) and size (100 documents) of comparing methods as well as similar ROUGE evaluation metrics (ROUGE-1, 2, and L). In the experimental part of this paper, summaries found by H1 were installed as a reference summary while summaries found by H2 had been installed as a benchmark method. Thus, H2 is used to measure out which one of all compared methods (BiDETS, set A, or set B) is closest to the human performance (H1).

E. Evaluation Tools

Most automatic text summarization research is evaluated and compared using the ROUGE tool to measure the quality of the system's summary. Citing such research studies isn't possible as there are too many to point out here. ROUGE stands for Recall-Oriented Understudy for Gisting. It presents measure sets to evaluate the system summaries; each set of metrics is suitable for specific kind of input type: very short single document summarization of 10 words, single document summarization of 100 words, and multi-document summarization of [10, 100, 200, 400] words. ROUGE-N (N=1 and 2) and L (L=longest common subsequence -LCS) single document measures are being used in this study; for more details about ROUGE the reader can refer to [8]. The ROUGE tool gives three types of scores: P=Precision, R=Recall, and F=F-measure for each metric (ROUGE-N, L, W, and so on). The F-measure computes the weighted harmonic mean of both P and R, see Equation (12):

$$F = \frac{1}{\left(alph \times \left(\frac{1}{P} \right) + \left(1 - alpha \right) \times \left(\frac{1}{R} \right) \right)} \quad (12)$$

Where $alpha$ is a parameter used to balance between both recall and precision.

For significance testing, to measure the performance of the system using each single score of n samples is a very tough job. Thus, the need to find a representative value replacing all n items emerged. ROUGE generalizes and expresses all values of (P, R, and F) results in single values (averages) respectively at a 95% confidence interval. Equations 14 and 15 declare how R and P are being calculated for evaluating text summarization system results respectively. For comparison purposes, some researchers are biased in selecting the Recall and Precision values such as [60, 65], and some of them biased behind selecting the F-measure as it reports a balance performance of both P and R of the system [66, 67]. In this study, the F-measure has been selected.

$$Recall = \frac{\left| System\ Summary \right| \cap \left| Human\ Summary \right|}{\left| Human\ Summary \right|} \quad (13)$$

$$Precision = \frac{\left| System\ Summary \right| \cap \left| Human\ Summary \right|}{\left| System\ Summary \right|} \quad (14)$$

V. Binary Differential Evolution based Text Summarization (BiDETS) Model

The proposed BiDETS model consists of two sub models: Differential Evolution for Features Selection (DEFS) model and Differential Evolution for Text Summarization (DETS) model. The term "Binary" is used here to refer to the current configuration of the system which was modulated into binary dimension space. Each sub model in the BiDETS acts as a separate model and runs independently of the other. The DEFS model is trained to extract the optimal weights of each feature. Then, the outputs of DEFS (the extracted weights) are directed as inputs to the second model. The DETS was designed to test the results of the trained model. Both models were trained and tested using the 10-fold approach (70% for training and 30% for testing). It is important to mention that all models were configured and prepared as discussed in Section 4: pre-processing all documents in the dataset, calculating the features, encoding the chromosome, configuring the DE's control parameters, and lastly assigning fitness function. Fig. 3 visualizes the whole BiDETS model.

Fig. 3 represent the Inputs and outputs of DEFS and DETS models, where F refers features and $W_i * F_i$ refers to a multiplication of the optimized weight W by the scored feature F.

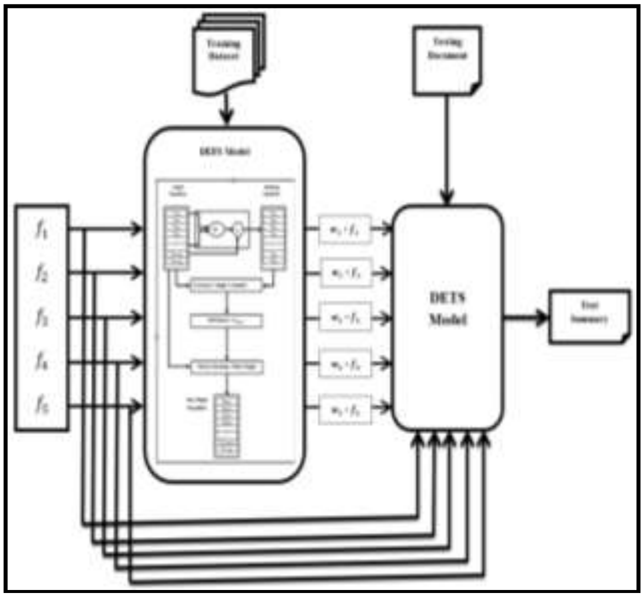

Fig. 3. Inputs and Outputs of DEFS and DETS Models.

A. DEFS Model

To generate a summary, the "features-based" Summarizer computes the feature scores of all sentences in the document. In the DEFS model, each chromosome deals with a separate document and controls the activation and deactivation of the corresponding set of features as shown in Fig. 2. The genes in the chromosome represent the five selected features. The DEFS model has been configured to operate in binary mode; if gene = 1 then the corresponding feature is active and will be included in the final score; otherwise the corresponding feature shall not be considered and will be excluded. In this way, the amount of 2^n probable chromosomes/solutions (where 2 represents the binary logic and n = 5 = number of selected features) can be obtained, and that is as follows. The model receives document

i(where 1≤i≤mr, mis a total number of the documents in the dataset)

, then starts its evolutionary search. For each initiated chromosome a corresponding summary has to be generated for this input document i. The model then triggers the evaluation toolkit ROUGE system and extracts "ROUGE-1 recall" value. Then it assigns this value as a fitness function to this current initiated chromosome. The DEFS continues generating an optimized multiple population and searching for the fittest solution. DEFS stops searching the space, similar to other evolutionary algorithms, when all the 100 iterations have been checked and then picks the highest fitness found. Once the fittest chromosome has been selected, the model stores its binary status into a binary-coded array of size 5×100, where 5th dimension (features) is and 100 is the length of the array (documents). Then, DEFS receives the next input (document i+1). When the system finishes searching all documents and fills the binary-coded array with all optimal solutions, then it computes the averages of all features in decimal format and stores it into a different array. The array is called real-coded array and it is of size m×n, where m=5 is the array dimension and n=10 is the total number of all runs. DEFS is now considered as finishing the first run out of 10. Then, the aforementioned steps are repeated until the real-coded array is filled and averages have been computed. These averages represent the target feature weights that a Summarizer designer is looking for. To this end, DEFS stops working and feeds those obtained weights as inputs to optimize the corresponding scored features of the DETS model. The DETS model was designed to test the results of the DEFS model as well as being installed as a final summarization application. Fig. 4 shows the obtained weights using DEFS ordered in descending manner for easy comparison.

The weights obtained by the DEFS model are represented in Fig. 3. These weights were organized in descending order for easy comparison. Each weight tells its importance and effect on the text. Firstly, one piece of literature showed that the title's frequency (TF) feature is a very important feature. It is well-known that, when people would like to edit an article, the sentences are designed to be close to the title. From this point of view, TF feature is very important to consider and DEFS supports this fact. Secondly, the sentence position (SP) feature is not less important that the TF as many experiments approved that the first and last sentence in the paragraph introduce and conclude the topic. The authors of this study have found that most of the selected document paragraphs are of short length (between two and three sentences). Then the authors followed to score sentences according to their sequenced appearance, and retain for the first sentence its importance. Thirdly, the thematic word gets an appreciated concern as it owns the events of the story. Take for example this article, the reader will notice that the terms "DEFS", "DETS", "chromosome" and "feature" are more frequently mentioned than the term "semantic" which was mentioned only once. These terms may represent the edges of this text. Thus, for the Summarizer it is good to include such feature. Fourthly, the sentence length feature also has a good effect as follows. In summarization the longest sentence may append with details which are irrelevant to the document topic; also short sentences lack informative information. For this reason, this feature is adjusted to enable the Summarizer to include a sentence of a suitable length. The importance of all mentioned features is ranged from score (0.80 to 0.99) except the last feature. Fifth, according to the definition of numerical feature, this feature is principally very important as it feeds the reader with facts and indications among the lines, for example the number of victims in an accident, the amount of stolen bank balances and so on. DEFS reports that the ND feature importance is acceptable but is the lowest one to weight. This reflects the ratio of presence (weight) of this feature which is (79%) in the documents. The authors have manually checked the texts and found that the presence of the numerical data is not so high. For real verification of these results, the weights are directed as input for the DETS model.

B. DETS Model

The DETS model is the summarization system which is designed with the selected features. The model scores features for each input document and generates a corresponding summary, see Equation (15).

$$Score\_F\left(S_{i\in[1..n]}\right)=\sum_{j=1}^{5}F_j\left(S_i\right) \tag{15}$$

where, $Score\_F()$ is a function that computes all j^{th} features F for all n document sentences. To optimize the scoring mechanism, DETS is fed with DEFS outputs to adjust the features. The weights can be set in the form of W= {w1, w2, w3, w4, w5}, where w refers to weight. Equation (17) shows how to combine the extracted weights with the scoring mechanism at Equation 16.

$$weighted\_Score\_F\left(S_{i\in[1..n]}\right)=\sum_{j=1}^{5}Score\_F_j\left(S_i\right)\times w_j \tag{16}$$

Where, w_{j} is j^{th} weight of the corresponded j^{th} feature.

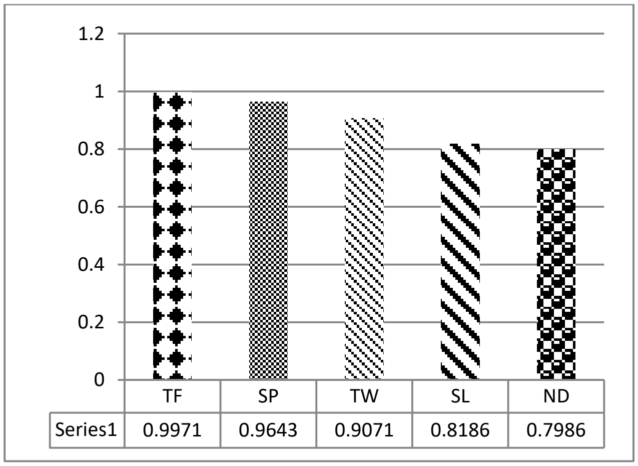

Fig. 4. Features Weights Extracted using DEFS Model.

Tables I, II, and III show a comparison of results between the three methods sets with the proposed method using ROUGE-1, ROUGE-2, and ROUGE-L at the 95%-confidence interval, respectively. The scores of the average recall (Avg_R), average precision (Avg_P), and average F-measure (Avg_F) are generalized at 95%-confidence interval. For each comparison the highest score result was styled in bold font format except the score of H2-H1. It is important to refer to the experimental results of GA published work, only the authors of this work had run ROUGE-1, and this study will depend and use this result through all comparative reviews. Fig. 5, 6, and 7 used to visualize results of the same Tables I, II, and III, respectively.

Two kinds of experiments were executed in this study: DEFS and DETS. The former is responsible for obtaining the adjusted weights of the features, while the latter is responsible for implementing the adjusted weights in a problem of text summarization.

TABLE I. DE, H2-H1, SET A AND B APPROACHES ASSESSMENTS USING ROUGE-1 RESULT AT THE 95%-CONFIDENCE INTERVAL

ROUGE	Method	Avg-R	Avg-P	Avg-F
1	H2-H1	0.51642	0.51656	0.51627
	B_Sys	0.40259	0.50244	0.43642
	W_Sys	0.06705	0.68331	0.1209
	DE	0.45610	0.52971	0.48495
	PSO	0.43028	0.47741	0.44669
	GA	0.45622	0.47685	0.46423

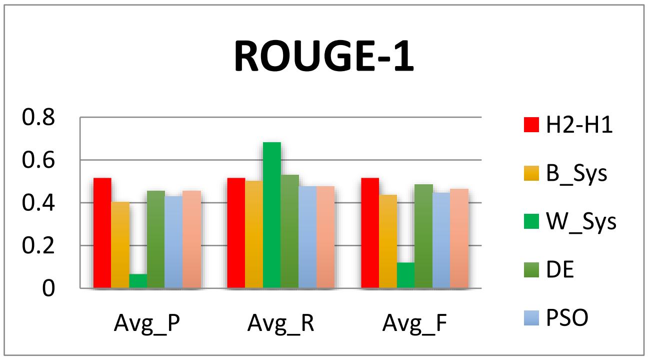

Fig. 5. DE, H2-H1, Set A, B and C Approaches Assessments using ROUGE-1 Avg_R, Avg-P, and Avg_F.

TABLE II. DE, H2-H1, SET A AND B APPROACHES ASSESSMENTS USING ROUGE-2 RESULT AT THE 95%-CONFIDENCE INTERVAL

ROUGE	Method	Avg-R	Avg-P	Avg-F
2	H2-H1	0.23394	0.23417	0.23395
	B_Sys	0.1842	0.24516	0.20417
	W_Sys	0.03417	0.38344	0.06204
	DE	0.24026	0.28416	0.25688
	PSO	0.18828	0.21622	0.19776
	GA	-	-	-

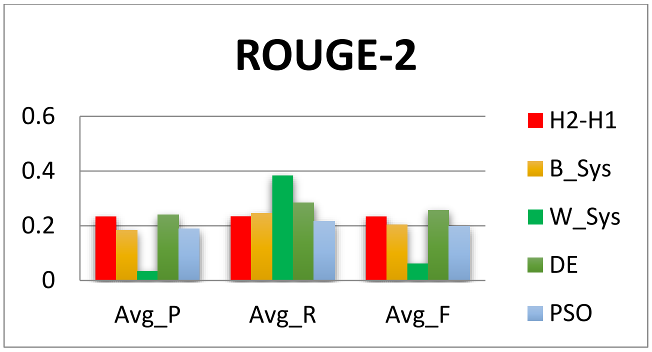

Fig. 6. DE, H2-H1, Set A and B Approaches Assessments using ROUGE-2 Avg_R, Avg-P, and Avg_F.

TABLE III. DE, H2-H1, SET A AND B APPROACHES ASSESSMENTS USING ROUGE-L RESULT AT THE 95%-CONFIDENCE INTERVAL

ROUGE	Method	Avg-R	Avg-P	Avg-F
L	H2-H1	0.48389	0.48400	0.48374
	B_Sys	0.37233	0.46677	0.40416
	W_Sys	0.06536	0.66374	0.11781
	DE	0.42000	0.48768	0.44645
	PSO	0.39674	0.44143	0.41221
	GA	-	-	-

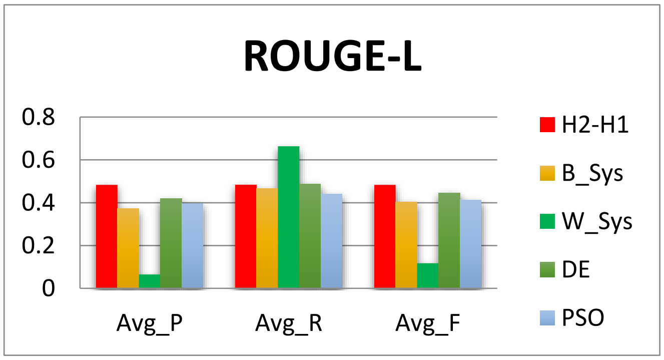

Fig. 7. DE, H2-H1, Set A and B Approaches Assessments using ROUGE-L Avg_R, Avg-P, and Avg_F.

The DETS model was designed to test and evaluate the performance of the DEFS model when performing feature selection for text summarization problems. The DETS model receives the weights of DEFS, as applies Equation 12 to score the sentences and generate a summary. The results showed that qualities of summaries generated using the (DE) are much better than the similar optimization techniques (set A: PSO and GA) and set C: best and worst system which participated at DUC 2002. The comparison of current extractive text summarization methods based on the DUC2002 has been investigated and reported as shown in Table IV.

Table IV demonstrates the comparison of some current extractive text summarization methods based on the DUC2002 dataset. On the basis of the findings generalized, the performance of the BiDETS model is 49% similar to human performance (52%) in ROUGE-1; 26% which is over the human performance (23%) using ROUGE-2; and lastly 45% similar to human performance (48%) using ROUGE-L.

TABLE IV. COMPARISON OF CURRENT EXTRACTIVE TEXT SUMMARIZATION METHODS BASED ON THE DUC2002

Method	Evaluation measure	Results

Evolutionary Optimization [68]	ROUGE-1, ROUGE-2, ROUGE-L	R-1 = 0.4990
		R-2 = 0.2548
		R-L = 0.4708

Hybrid Machine Learning [69]	ROUGE-1	R-1 = 0.3862
Statistical and linguistic [70]	F-measure	F-measure = 25.4 %

IR using event graphs[71]	ROUGE-1 ROUGE-2	R-1 = 0.415
		R-2 = 0.116

Genetic operators and guided local search[72]	ROUGE-1 ROUGE-2	R-1 = 0.48280
		R-2 = 0.22840

Graph-based text summarization [73]	ROUGE-1 ROUGE-2	R-1 = 0.485
		R-2 = 0.230

SumCombine [74]	ROUGE-1 ROUGE-2	R-1 = 0.3823
		R-2 = 0.0946

Self-Organized and DE [17]		20% in R-1 05% in R-2

Discrete DE[30]	Recall
	Precision
	F-measure
		Recall = 0.74
		Precision = 0.32
		F-measure = 0.44

BiDETS		52% in R-1 23% in R-2 48% in R-L

This study has approved two contributions: firstly; studying the importance of the text features fairly could lead to producing a good summary. The obtained weights from the trained model were used to tune the feature scores. These tuned scores have a noted effect on the selection procedure of the most significant sentences to be involved in the summary. Secondly, developing a robust feature scoring mechanism is independent of the means of innovating novel features with different structure or proposing complex features. This study had experimentally approved that adjusting the weights of simple features could outperform systems that are either enriched with complex features or run with a high number of features. In contrast of testing phase to other methods classified in set A and B, the proposed Differential Evolution model demonstrated good performance.

VI. CONCLUSION AND FUTURE WORK

In this paper, the evolutionary algorithm "Differential Evolution" was utilized to optimize feature weights for a text summarization problem. The BiDETS scheme was trained and tested with a 100 documents gathered from the DUC2002 dataset. The model had been compared to three sets of selected benchmarks of different types. In addition, the DE employed simple calculated features compared with features presented in PSO and GA models. The PSO method assigned five features that differed in their structure: complex and simple; while the GA was designed with eight simple features. The DE in this study was deployed with five simple features and it was able to extract optimal weights that enabled the BiDETS model to generate summaries that were more qualified than other optimization algorithms. The BiDETS concludes that feeding the proposed systems with many, or complex features, instead of using the available features, may not lead to the best summaries. Only optimizing the weights of the features may result in generating more qualified summaries as well as employing a robust evolutionary algorithm. The ROUGE tool kit was used to evaluate the system summaries in 95% confidence intervals and extracted results using the average recall, precision, and F-measure of ROUGE-1, 2 and L. The F-measure was chosen as a selection criterion as it balances both the recall and the precision of the system's results. Results showed that the proposed BiDETS model outperformed all methods in terms of F-measure evaluation.

The main contribution of this research is generate a short text for the input document; this short text should represent and contain the important information in the document. A sentence extraction is one of the main techniques that is used to generate such a short text. This research is concerned about sentence extraction integrate an intelligent evolutionary algorithm by producing optimal weights of the selected features for generating a high quality summary. In addition, the proposed method enhanced the search performance of the evolutionary algorithm and obtained more qualified results compared to its traditional versions. In contrast, It is worth mentioning that, the summary measure basically used to score (document, query) similarity for large numbers of web pages. So, computing the similarity measure is disadvantageous to single document text summarization.

For future works, we assume that compressibility may prove to be a valuable metric to research the efficiency of automated summarization systems and even perhaps for text identification if for instance, any authors are found to be reliably compressible. In addition, this study opens a new trend for encouraging researchers to involve and implement other evolutionary algorithms such as Bee Colony Optimization (BCO) [75] and Ant Colony Optimization (ACO) [76] to draw more significant comparisons of optimization based Feature Selection text summarization issue. In addition and to the optimal of the author's finding, the literature presented that algorithms such as ACO and BCO have not been presented yet to tackle general text summarization issues of both "Single" and "Multi-Document" Summarization. A second future work is to integrate the results of this study with a technique of "diversity" in summarization. This is to enable the DE selecting more diverse sentences to increase the result quality of the summarization.

ACKNOWLEDGMENT

This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia under grant No. (J:218-830-1441). The authors, therefore, gratefully acknowledge DSR for technical and financial support.

REFERENCES

[1] G. C. V. Vilca and M. A. S. Cabezudo, "A study of abstractive summarization using semantic representations and discourse level information," in International Conference on Text, Speech, and Dialogue, 2017, pp. 482-490.

[2] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," Artificial Intelligence Review, vol. 47, pp. 1-66, 2017.

[3] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence: MIT Press, 1992.

[4] R. C. E. J. Kennedy, and Y. Shi, Swarm Intelligence. London: Morgan Kaufmann Publishers, 2001.

[5] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," J. of Global Optimization, vol. 11, pp. 341-359, 1997.

[6] D. Harman and P. Over, "The DUC summarization evaluations," in Proceedings of the second international conference on Human Language Technology Research, 2002, pp. 44-51.

[7] DUC, "The Document Understanding Conference (DUC).".

[8] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of summaries," presented at the Proc. ACL workshop on Text Summarization Branches Out, 2004.

[9] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in Evolutionary Computation, 2004. CEC2004. Congress on, 2004, pp. 1980-1987 Vol.2.

[10] H. Xingshi, Qingqing, Zhang, Na, Sun, Yan, Dong, "Feature Selection with Discrete Binary Differential Evolution," in Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on, 2009, pp. 327-330.

[11] R. N. Khushaba, Al-Ani, A., Al-Jumaily, A., "Differential evolution based feature subset selection," in Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, 2008, pp. 1-4.

[12] N. Nazari and M. Mahdavi, "A survey on automatic text summarization," Journal of AI and Data Mining, vol. 7, pp. 121-135, 2019.

[13] M. Kirmani, N. M. Hakak, M. Mohd, and M. Mohd, "Hybrid text summarization: a survey," in Soft Computing: Theories and Applications, ed: Springer, 2019, pp. 63-73.

[14] A. Kumar and A. Sharma, "Systematic literature review of fuzzy logic based text summarization," Iranian Journal of Fuzzy Systems, vol. 16, pp. 45-59, 2019.

[15] M. A. Mosa, A. S. Anwar, and A. Hamouda, "A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms," Knowledge-Based Systems, vol. 163, pp. 518-532, 2019.

[16] D. Suleiman and A. A. Awajan, "Deep Learning Based Extractive Text Summarization: Approaches, Datasets and Evaluation Measures," in 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2019, pp. 204-210.

[17] N. Saini, S. Saha, A. Jangra, and P. Bhattacharyya, "Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm," Knowledge-Based Systems, vol. 164, pp. 45-67, 2019.

[18] S. Gupta and S. Gupta, "Abstractive summarization: An overview of the state of the art," Expert Systems with Applications, vol. 121, pp. 49-65, 2019.

[19] H. Lin and V. Ng, "Abstractive summarization: A survey of the state of the art," in Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 9815-9822.

[20] J. Tandel, K. Mistree, and P. Shah, "A review on neural network based abstractive text summarization models," in 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), 2019, pp. 1-4.

[21] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh, "A comparative study of summarization algorithms applied to legal case judgments," in European Conference on Information Retrieval, 2019, pp. 413-428.

[22] A. Kanapala, S. Pal, and R. Pamula, "Text summarization from legal documents: a survey," Artificial Intelligence Review, vol. 51, pp. 371-402, 2019.

[23] F. Jacquenet, M. Bernard, and C. Largeron, "Meeting summarization, A challenge for deep learning," in International Work-Conference on Artificial Neural Networks, 2019, pp. 644-655.

[24] L. Ermakova, J. V. Cossu, and J. Mothe, "A survey on evaluation of summarization methods," Information Processing & Management, vol. 56, pp. 1794-1814, 2019.

[25] A. Mahajani, V. Pandya, I. Maria, and D. Sharma, "A comprehensive survey on extractive and abstractive techniques for text summarization," in Ambient Communications and Computer Systems, ed: Springer, 2019, pp. 339-351.

[26] M. A. Mosa, "A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining," Applied Soft Computing, vol. 90, p. 106189, 2020.

[27] M. A. Mosa, A. Hamouda, and M. Marei, "Graph coloring and ACO based summarization for social networks," Expert Systems with Applications, vol. 74, pp. 115-126, 2017.

[28] M. A. Mosa, "Real-time data text mining based on Gravitational Search Algorithm," Expert Systems with Applications, vol. 137, pp. 117-129, 2019.

[29] L. Chen and M. Le Nguyen, "Sentence Selective Neural Extractive Summarization with Reinforcement Learning," in 2019 11th International Conference on Knowledge and Systems Engineering (KSE), 2019, pp. 1-5.

[30] S. Karwa and N. Chatterjee, "Discrete differential evolution for text summarization," in 2014 International Conference on Information Technology, 2014, pp. 129-133.

[31] L.-y. C. Chung-jui Tu, Jun-yang Chang, Cheng-hong Yang, "Feature Selection using PSO-SVM," IAENG International Journal of Computer Science, vol. 33, pp. 138-143, 2006.

[32] Y. Liu, Qin, Zheng, Xu, Zenglin, He, Xingshi, "Feature Selection with Particle Swarms," in Computational and Information Science. vol. 3314, J. Zhang, J.-H. He, and Y. Fu, Eds., ed: Springer Berlin / Heidelberg, 2005, pp. 425-430.

[33] J. Rojas-Simón, Y. Ledeneva, and R. A. García-Hernández, "Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm," Expert Systems with Applications, p. 113827, 2020.

[34] A. Kiani and M. R. Akbarzadeh, "Automatic Text Summarization Using Hybrid Fuzzy GA-GP," in Fuzzy Systems, 2006 IEEE International Conference on, 2006, pp. 977-983.

[35] R. Ferreira, L. de Souza Cabral, R. D. Lins, G. P. e Silva, F. Freitas, G. D. Cavalcanti, et al., "Assessing sentence scoring techniques for extractive text summarization," Expert systems with applications, vol. 40, pp. 5755-5764, 2013.

[36] M. S. Binwahlan, Salim, N., & Suanmali, L., "Swarm based features selection for text summarization," International Journal of Computer Science and Network Security IJCSNS, vol. 9, pp. 175-179, 2009b.

[37] D. Liu, Y. He, D. Ji, and H. Yang, "Genetic algorithm based multi-document summarization," in Pacific Rim International Conference on Artificial Intelligence, 2006, pp. 1140-1144.

[38] A. Zamuda and E. Lloret, "Optimizing Data-Driven Models for Summarization as Parallel Tasks," Journal of Computational Science, p. 101101, 2020.

[39] J. Y. Yeh, H. R. Ke, W. P. Yang, and I. H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," Information Processing & Management, vol. 41, pp. 75-95, Jan 2005.

[40] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," Computer Speech and Language, vol. 23, pp. 126-144, Jan 2009.

[41] G. Pampara, A. P. Engelbrecht, and N. Franken, "Binary Differential Evolution," in Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006, pp. 1873-1879.

[42] R. M. Alguliev and R. M. Aliguliyev, "Evolutionary Algorithm for Extractive Text Summarization.," Intelligent Information Management, vol. 1, pp. 128-138, 2009.

[43] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization," Knowledge-Based Systems.

[44] Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," Applied Soft Computing, vol. 11, pp. 1363-1379, 2011.

[45] R. M. Aliguliyev, "Clustering of document collection – A weighting approach," Expert Systems with Applications, vol. 36, pp. 7904-7916, 2009.

[46] G. Liu, Y. Li, X. Nie, and H. Zheng, "A novel clustering-based differential evolution with 2 multi-parent crossovers for global optimization," Applied Soft Computing, vol. 12, pp. 663-681, 2012.

[47] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," Appl. Soft Comput., vol. 11, pp. 1679-1696, 2011.

[48] M. L. Marina Litvak, Menahem Friedman, "A new approach to improving multilingual summarization using a genetic algorithm," presented at the Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 2010.

[49] S. S. C. N. Satoshi, M. Murata, K. Uchimoto, M. Utiyama, and H. Isahara, "Sentence Extraction System Assembling Multiple Evidence," in 2nd NTCIR Workshop, 2001, pp. 319--324.

[50] H. P. Luhn, "The automatic creation of literature abstracts," IBM J. Res. Dev., vol. 2, pp. 159-165, 1958.

[51] H. P. Edmundson, "New Methods in Automatic Extracting," J. ACM, vol. 16, pp. 264-285, 1969.

[52] J. L. Neto, Santos, A. D., Kaestner, C. A. A., Freitas, A. A., "Generating text summaries through the relative importance of topics," Advances in Artificial Intelligence, vol. 1952, pp. 300-309, 2000.

[53] P. B. Baxendale, "Machine-made index for technical literature: an experiment," IBM J. Res. Dev., vol. 2, pp. 354-361, 1958.

[54] S. Rahnamayan and H. Tizhoosh, "Differential Evolution Via Exploiting Opposite Populations Oppositional Concepts in Computational Intelligence." vol. 155, H. Tizhoosh and M. Ventresca, Eds., ed: Springer Berlin / Heidelberg, 2008, pp. 143-160.

[55] L. Junhong and L. Jouni, "A fuzzy adaptive differential evolution algorithm," in TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, 2002, pp. 606-611 vol.1.

[56] M. M. Ali, A. T\, \#246, and rn, "Population set-based global optimization algorithms: some modifications and numerical studies," Comput. Oper. Res., vol. 31, pp. 1703-1725, 2004.

[57] R. Thangaraj, M. Pant, and A. Abraham, "New mutation schemes for differential evolution algorithm and their application to the optimization of directional over-current relay settings," Applied Mathematics and Computation, vol. 216, pp. 532-544, 2010.

[58] L. Chang-Yong and Y. Xin, "Evolutionary programming using mutations based on the Levy probability distribution," Evolutionary Computation, IEEE Transactions on, vol. 8, pp. 1-13, 2004.

[59] Y. Xin, L. Yong, and L. Guangming, "Evolutionary programming made faster," Evolutionary Computation, IEEE Transactions on, vol. 3, pp. 82-102, 1999.

[60] M. S. Binwahlan, Salim, N., Suanmali, L, "Swarm Based Text Summarization," in Computer Science and Information Technology - Spring Conference, 2009. IACSITSC '09. International Association of, 2009, pp. 145-150.

[61] M. F. Porter, "An algorithm for suffix stripping," in Readings in information retrieval, ed: Morgan Kaufmann Publishers Inc., 1997, pp. 313-316.

[62] L. Suanmali, N. Salim, and M. S. Binwahlan, "Genetic Algorithm Based for Sentence Extraction in Text Summarization," International Journal of Innovative Computing, vol. 1, 2011.

[63] S. H. Finley and S. M. Harabagiu, "Generating single and multi-document summaries with GISTEXTER," in Proceedings of the workshop on text summarization, Philadelphia, PA, USA, 2002.

[64] D. Zajic, B. Dorr, and R. Schwartz, "Automatic headline generation for newspaper stories," in Proceedings of the workshop on text summarization, Philadelphia, Pennsylvania, USA, 2002.

[65] L.-C. Yu, C.-H. Wu, and F.-L. Jang, "Psychiatric document retrieval using a discourse-aware model," Artificial Intelligence, vol. 173, pp. 817-829, 2009.

[66] R. M. Alguliev and R. M. Aliguliyev, "Experimental investigating the F-measure as similarity measure for automatic text summarization," Applied and Computational Mathematics, vol. 6, pp. 278-287, 2007.

[67] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," Artificial Intelligence.

[68] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," Expert Systems with Applications, vol. 40, pp. 1675-1689, 2013.

[69] M. A. Fattah, "A hybrid machine learning model for multi-document summarization," Applied intelligence, vol. 40, pp. 592-600, 2014.

[70] R. Ferreira, L. de Souza Cabral, F. Freitas, R. D. Lins, G. de França Silva, S. J. Simske, et al., "A multi-document summarization system based on statistics and linguistic treatment," Expert Systems with Applications, vol. 41, pp. 5780-5787, 2014.

[71] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," Expert systems with applications, vol. 41, pp. 6904-6916, 2014.

[72] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos, and E. León, "Extractive single-document summarization based on genetic operators and guided local search," Expert Systems with Applications, vol. 41, pp. 4158-4169, 2014.

[73] D. Parveen and M. Strube, "Integrating importance, non-redundancy and coherence in graph-based extractive summarization," in Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[74] K. Hong, M. Marcus, and A. Nenkova, "System combination for multi-document summarization," in Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 107-117.

[75] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Kayseri, Turkey2005.

[76] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," Trans. Sys. Man Cyber. Part B, vol. 26, pp. 29-41, 1996.