Swarm and Evolutionary Computation ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Regular Paper

Extracting easy to understand summary using differential evolution algorithm

K. Nandhini *, S.R. Balasundaram

Department of Computer Applications, National Institute of Technology, Thiruchirappalli, India

ARTICLE INFO

Article history:
Received 22 January 2013
Received in revised form
26 June 2013
Accepted 13 December 2013

Keywords:
Extractive summarization
Differential evolution
Reading difficulties
Evolutionary computing
Combinatorial optimization

ABSTRACT

This paper describes an optimization method based on differential evolution algorithm and its novel application to extract easy to understand summary for improving text readability. The idea is to improve the readability of the given text for reading difficulties using assistive summary. In order to extract easy to understand summary from the given text, an improved differential evolution algorithm is proposed. A new chromosome representation that considers ordering and similarity for extracting cohesive summary. Also a modified crossover operator and mutation operator are designed to generate potential offspring. The application of differential evolution algorithm for maximizing the average similarity and informative score in the candidate summary sentences is proposed. We applied the proposed algorithm in a corpus of educational text from ESL text books and in graded text. The results show that the summary generated using Differential Evolution algorithm performs better in accuracy, readability and lexical cohesion than existing techniques. The task based evaluation done by target audience also favors the significant effect of assistive summary in improving readability.

1. Introduction

Over the past five decades, various researches on text summarization methods have been proposed and evaluated. The main objective of text summarization is automatic selection of text passages that represent the whole document. There are two main approaches in the task of summarization-extraction and abstraction [1]. Extraction involves concatenating extracts taken from the corpus into a summary, whereas abstraction involves generating novel sentences from information extracted from the corpus. The extractive summarization techniques can be further classified into two groups: the supervised techniques that rely on pre-existing document-summary pairs, and the unsupervised techniques, based on properties and heuristics derived from the text. Supervised extractive summarization techniques treat the summarization task as a two-class classification problem at the sentence level [2,3]. Many unsupervised methods have been developed for document summarization by exploiting different features and relationships of the sentences [4,5]. Early research on text summarization exploits various features such as word frequency [6], sentence [7], cue phrases [8], sentence length and upper case letter [2], TF-IDF [9], etc. Nowadays, corpus-based approaches play an important role in text summarization [2,10]. By exploiting technologies of machine learning, it becomes possible to learn rules from a corpus of documents and their corresponding summaries. The major advantage is that corpus-based approaches are easy to implement.

The design and evaluation of summarizing systems have to be related to the three classes of context factor namely input factors, purpose factors and output factors [11]. Most of the summarization systems available are designed for generic. There exist various specialized versions of summarization for disabled, such as blind, [12] deaf [13] and so on. The targeted audience and the purpose mainly determine the system design and evaluation [14]. Our targeted audience are learners with reading difficulties those capable of decoding but find hard in understanding the content better. Many of these students have difficulty in finding main ideas and important supporting details [1]. Failure to employ appropriate learning strategies is often a critical component of learning disabilities [15]. The generalized deficits in reading comprehension of many students with learning disabilities suggest the importance of systematic instruction in learning strategies. It is evident that the effect of summarization strategy in comprehending the text for reading difficulties is significant [16]. The purpose of summary is to aid the reading difficulties in improving the text readability which in turn helps in understanding the content better.

Much of the summarization work done so far has not referred to summary use, which mainly decides the system design. Our objective is to design a system for summary extraction that contains important, readable and cohesive sentences. To solve this problem, we propose an algorithm that extracts and order the sentences simultaneously, maximizes the informative, readable, cohesive score of the summary. The proposed algorithm efficiently searches for the best combination of sentences using differential evolutionary algorithm.

* Corresponding author.
E-mail address: nandhinikumaresh@yahoo.co.in (K. Nandhini).

2210-6502/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.swevo.2013.12.004

Our contribution in this paper are as follows:

1. Proposing features for enhancing cohesion, readability and informative score of summary sentences.
2. Proposing a new summarization method for incorporating various features to produce easy to understand summary for reading difficulties.
3. Performing both intrinsic and extrinsic evaluation to validate the effectiveness of the proposed method.

The rest of the paper is organized as follows: Section 2 deals with related works in text summarization. Section 3 explains the summary extraction process followed by differential evolution in combinatorial optimization process. Final section discusses the results with existing techniques and task based evaluation.

2. Related works

There are two approaches for document summarization namely supervised [17,3] and unsupervised [5]. The supervised approaches treat document summarization as a classification which requires training samples to classify sentences as summary or not. Many unsupervised methods have been developed for document summarization by exploiting different features and relationships of the sentences, such as clustering of sentences [5], the hidden topics in the documents [18], and graphs based on the similarity of sentences [19,20]. The graph-based extractive summarization algorithms identify the most important sentences in a text based on information exclusively drawn from the text rather than depending on training samples. The graph-based methods are fully unsupervised, and rely on the given texts to derive an extractive summary [23]. Several developments of summarization techniques based on graphs are reported in the literature. Salton et al. [21] considered the paragraphs as nodes, which are interconnected according to a similarity measure based on the number of words they share. Mani and Bloedorn, [22] represent the instances of terms as nodes, which are connected by cohesion relations such as proximity, repetition, synonymy and co-reference. In Mihalcea′s work, [23] ranks are given by recommendation algorithms such as PageRank [24] and HITS [25] for sentence extraction. In extractive document summarization, finding an optimal summary can be viewed as a combinatorial optimization problem which is NP-hard to solve. The idea of optimizing summarization was mentioned in [26]. They represented documents in a two dimensional space of textual and conceptual units with an associated mapping between them, and proposed a formal model that simultaneously selected important text units and minimized information overlap between them. Graph based problems are mainly NP-complete, meaning that a guaranteed optimal solution cannot be reached in polynomial time. Because a large number of problems in science and engineering can be formulated as graph layout problems, a variety of methods have been proposed for addressing them. These methods are mainly heuristic in nature and based on graph-theoretic concepts. The best graph-theoretic heuristic algorithms can produce good-quality solutions in a short time, but, of course, they do not guarantee the optimality of the solutions obtained, and the solutions may be far from ideal. Meta-heuristic approaches are popular alternative to classical optimization techniques in a variety of domains. Different meta-heuristics such as Simulated Annealing (SA) [27], Tabu Search (TS) [28], Genetic Algorithm (GA) [29] and Ant Colony (AC) [30] are currently used to solve the NP-hard problems. In this paper, we focused on the application of DE to solve extractive summarization.

The Differential Evolution (DE) algorithm was first proposed for optimization with continuous variables [31] and has been applied with success in many combinatorial optimization problems like job shop scheduling [32]. The survey of Discrete evolution is given in [33]. However various problems such as traveling salesman problem(TSP), involve integer optimization variables that are symbolic, not representing any numeric quantity. When applying the differential mutation to problems with symbolic variables, the differential vectors do not generate feasible solutions and do not represent any meaning direction due to arbitrary labeling. Aiming at the discrete problems, novel discrete DE approaches have been proposed in recent literature to solve combinatorial optimization problems [34]. The idea of this summary extraction using TSP is derived from [35] TSP and TSP using DE was derived from [36] with a main difference, the order of sentence is a major criterion for improving cohesion while extracting summary sentences. In this paper, a modified crossover and modified mutation is proposed for Differential Evolution algorithm.

3. Proposed methodology

Let the document D is composed by a set of sentences $D = \{S_1, S_2, S_3, \ldots S_n\}$ where each $S_i = \{t_1, t_2, t_3, \ldots t_m\}$ be all distinct terms occurring in a sentence of document D, 'n' represents the number of sentences and 'm' represents the number of terms. The first step in the proposed methodology is preprocessing which is explained in Algorithm 2 and features are extracted at various phases of preprocessing.

Algorithm 1. Proposed methodology.

1: **procedure** SUMMARY EXTRACTION(.txt file)
2: Preprocessing ▹ Sentence Segmentation, Tokenizer, Parts of speech tagging, Stemming
3: Feature Extraction ▹ Readability,Informative and Cohesive Features
4: Representing Data in VSM ▹ Vector Space Model
5: Calculating Informative Score ▹ Sum of Weighted Score of Features
6: Applying Differential Evolution algorithm ▹ Target Population
7: Finding optimal sentence combination ▹ Combinatorial Optimization
8: **return** *Candidate Sentences* ▹ Summary sentences
9: **end Procedure**

3.1. Preprocessing

During preprocessing, each word of the input document is written on a separate file. Each module either performs certain preprocessing tasks such as segmentation, tokenizing or attaches additional features such as parts of speech tags to the input texts. The preprocessing modules are as follows:

1. *Sentence segmentation*: Reads the text and segment it into sentences.
2. *Tokenizer*: Reads the sentences and outputs tokenized texts.
3. *Parts-of-speech tagger*: Reads tokenized texts and outputs part of speech tagged texts.
4. *Stop word removal*: Removes less important and meaning less words such as a, the, is etc.,
5. *Syllable counter*: Counts the occurrence of syllables in each word.
6. *Stemmer*: Finds all root forms of each input text.
7. *tf-idf calculator*: Calculates the tf-idf weights for each input token.

Please cite this article as: K. Nandhini, S.R. Balasundaram, Extracting easy to understand summary using differential evolution algorithm, Swarm and Evolutionary Computation (2014), http://dx.doi.org/10.1016/j.swevo.2013.12.004

3.2. Feature extraction

The feature extraction depends mainly on the purpose and the target audience of the text summarization. The features have to represent the text importance, learner difficulty factor and the motivation of the application. To identify the important sentences from the text, features like sentence position [7], centrality [37], title similarity [2], and keyword occurrences [8] are considered. The use of summarization is to aid the reading difficulties. As readability is one of the prime factors to be considered, we used features used by the Flesh-Kincaid [38] and FOG [39] metrics such as average words per sentence, average syllables per word, and percentage of words in the document with 3+ syllables. Sentences with more entities are more difficult for users to semantically encode due to working memory limitations [40]. So, the feature noun occurrences measure the average number of entities per sentence, the reader has to keep in mind to understand the sentence. As the learner may have problem in mapping trigger words [41], average trigger words per sentences are considered as another feature. For coherent sentences cosine similarity between the sentences is considered as another feature in extraction. The objective is to extract the important, readable and coherent sentences as summary.

Algorithm 2. Preprocessing and feature extraction.

```
procedure FEATURE EXTRACTION(.txt file)
  while(! EOF) do
    CALL Sentence Segmenter()
    extract features F1, F3  ▹ Sentence Position,Avg sentence length
for all Si do           ▹ For all Sentences in the text
    CALL Tokenizer()
    extract features F2          ▹ Trigger words
end For
for all Ti do           ▹ For all Tokens in Sentences
    CALL POS tagger()
    CALL Stopword removal()
    CALL syllable-counter()
    extract features F5, F6  ▹ Polysyllabic words, Noun occurrences
  end for
  for all CWi do           ▹ For all Content words in Tokens
    CALL Stemming()
    CALL tf-idf calculator()
    extract features F8, F9       ▹ Term Frequency, Keywords
end for
    extract features F4, F7      ▹ Title similarity, Centrality
  end while
  return Features              ▹ Features={F1...F9}
end procedure
```

3.2.1. Informative features

Sentence position: Sentence position is an important factor to decide sentence importance in the document [7,8]. The significance of the position of sentence plays a vital role in various domains accordingly. For news domain, sentences at the beginning of the text carry important information. In the case of scientific documents important information lies mostly, in the abstract and in the conclusion [2]. The important sentences to be included in the summary are usually located in a particular position. In the case of educational text, boundaries carry important information. The score for the sentences is calculated as follows:

$$Sentence\ position = Max\left(\frac{1}{i}, \frac{1}{n-i+1}\right) \tag{1}$$

where "'n"' is the total number of sentences in paragraph and "'i"' is the location of the sentence in the paragraph.

Title to sentence similarity: Titles contain the group of words that give important clues about concepts of the text [2,8]. If the sentence has higher intersection with the title words then the score for the sentence is calculated as follows:

$$Title\ similarity = \frac{|Words\ in\ title\ \cap\ words\ in\ S_{(}i)|}{|Words\ in\ title\ \cup\ words\ in\ S_{(}i)|} \tag{2}$$

Centrality: The centrality of the sentence implies its similarity to other sentences [37]. If a sentence has higher centrality, then those sentences are the best candidates to be included in the summary and their score is calculated as follows:

$$Centrality = \frac{|Words\ in\ S_i \cap Words\ in\ S_{(}n-i)|}{|Words\ in\ S_i \cup\ Words\ in\ S_{(}n-i)|} \tag{3}$$

Cue phrases: The sentences contain cue phrases like "defined as", "called", "significant", "means", "important" contain important definitions that are to be added in the important class category [8]. The importance of the word is determined by its term frequency which is calculated as follows: Tf-isf is a statistical technique used to evaluate how important a word is to a document [42,9]. The term frequency in the given document is the number of times a given term appears in the document:

$$Tf_i = \frac{T_i}{\sum_{i=1}^{n} T_k} \tag{4}$$

where T_i is the number of occurrences of the term and T_k is the sum of occurrences of all the terms in the document. The inverse sentence frequency is a measure of the importance of the term:

$$isf = \log\left(\frac{N}{n_i}\right) \tag{5}$$

where N is the number of sentences in the document and n is the number of sentences containing the significant term. The corresponding weight is therefore computed as

$$Score = Tf * isf \tag{6}$$

3.2.2. Readability features

As the targeted audience are having reading difficulties, features that extract easy to understand sentences help them in reading and in turn understanding the text better.

Sentence length: The major reason for having difficulty in text comprehension is due to the individual's working memory problem. The reader was unable to integrate the semantic of words present in the sentences. Longer the sentence, the learner may fail to integrate it completely. Usually longer sentences carry important messages [2]. But accessibility issues favor shorter ones. Learners having problem in working memory may not be benefited even though the longer sentences are important. Due to short term memory and sequencing difficulties, students can lose their way in comprehending texts because they are not retaining important facts and therefore do not fully understand the text.

$$Average\ sentence\ length = \frac{\#(Len(S_i))}{N} \tag{7}$$

Word length: Sentences containing longer and infrequent (can be identified using term frequency) words are hard to follow, that should not include proper nouns or combination of easy words (hyphenated). Sentences having higher number of hard words are considered as difficult sentences [43], The percentage of hard words in the text can be calculated as follows:

$$Percentage\ hard\ words = \frac{\#(HW(S_i))}{Len(S_i)} \tag{8}$$

Please cite this article as: K. Nandhini, S.R. Balasundaram, Extracting easy to understand summary using differential evolution algorithm, Swarm and Evolutionary Computation (2014), http://dx.doi.org/10.1016/j.swevo.2013.12.004

Trigger words: Occurrences of more number of trigger words create distraction to the dyslexic reader [41]. Sentences having higher number of trigger words are considered as difficult sentences for the audience [44]. The percentage of trigger words in the text can be calculated as follows:

$$\text{Percentage trigger words} = \frac{\#(TW(S_i))}{Len(S_i)} \tag{9}$$

Polysyllabic words: Words that are harder to decode will increase the difficulty level of comprehension. The average number of syllables and the average sentence length have been used to determine the reading difficult of the text as in SMOG (Simple Measure Of Gobbledegook). Hence the number of syllables in a word is also one of the measures of word difficulty. The percentage of polysyllabic words in the text can be calculated as follows:

$$\text{Percentage polysyllabic words} = \frac{\#(PSYW(S_i))}{Len(S_i)} \tag{10}$$

Noun occurrences: The number of nouns introduced in a text relates to the working memory burden on the target audience. The noun (extracted using POS tagging) counts in individual sentences also decides its easiness or difficult nature. Usually the sentence that contains more proper nouns is an important one and it is most probably included in the summary. But for the readers with difficulty, more nouns create confusion.

$$\text{Percentage noun occurrences} = \frac{\#(Noun(S_i))}{Len(S_i)} \tag{11}$$

3.3. Data representation

The sentences in a document are represented using the Vector Space Model (VSM), as a vector in the term-space. In this model, each sentence S_i is located as a point in a m-dimensional vector space, $S_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, $i = 1, \ldots, n$, where the w_{ik} is the weight of the kth term in ith sentence. Each component of such a vector reflects a term connected with the given document. The value of each component depends on the degree of relationship between its associated term and the respective sentence can be represented using term frequency (tf-isf).

3.4. Finding feature weight

Mathematical Regression is a good model to estimate the text feature weights [45,46]. In this model a mathematical function relates output to input. The feature parameters are used as independent input variables and the corresponding reference summary sentences selection are specified as ‘1’ act as output variables. In matrix notation, we can represent regression as follows: Here, X is the input matrix (feature parameters). Y is the output vector. W is the linear statistical model of the system (weights $W_1 \ldots W_n$) in (12).

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots x_{19} \\ x_{21} & x_{22} & x_{23} & \cdots x_{29} \\ . & & & \\ . & & & \\ . & & & \\ x_{n1} & x_{n2} & x_{n3} & \cdots x_{n9} \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ . \\ . \\ . \\ W_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} \tag{12}$$

where X_{ij} is the value of jth feature in ith sentence and W_j is the weight of feature “j”. The sentence score can be calculated as a linear combination of the feature vector and its corresponding weight.

where $[Y]$ is the output vector, $[X]$ is the input matrix (feature parameters), $[W]$ is the linear statistical model of the system (the weights $w1$, $w2$,…$w10$). The feature weights are calculated using regression model. In order to find the suitable weights for the individual features, a set of documents (features) along with class (gold summary) are considered.

3.5. Calculating sentence score

The sentence score can be calculated as linear combination of the feature vector and its corresponding weight:

$$Score = \sum_{i=1}^{n} W_i * F_i \tag{13}$$

Algorithm 3. Sentence score.

```
procedure SENTENCE SELECTION {s} ⊆ {S}
    Input: {F},T            ▹ Feature vector,Similarity Threshold
    Output: {s}             ▹ candidate sentences
    for i←1, n do
        S[i] = Σ_{j=1}^{9} W_j*F_j      ▹ Calculate sentence score
    end for
end procedure
```

3.6. Calculating similarity score

Similarity measures play an increasingly important role in text-related research and applications such as text summarization, document clustering, and information retrieval. It is possible to compute pairwise similarity between pairs of texts, based on coincidence in the term assignment to the respective terms. A variety of similarity or distance measures have been proposed and widely applied, such as euclidean distance, Jaccard coefficient, cosine similarity and so on. The cosine similarity has been one of the most popular document similarity measures due to its sensitivity to document vector patterns. The cosine measure computes the cosine of the angle between two feature vectors and is used in a very large and sparse text. The cosine similarity between two sentences can be calculated as follows:

$$Sim(S_i, S_l) = \text{Cos}(S_i, S_l) = \frac{S_i . S_l}{\| S_i | . | S_l \|} \tag{14}$$

$$Sim(S_i, S_l) = \frac{\sum_{j=1}^{m} W_{ij} * W_{lj}}{\sqrt{\sum_{j=1}^{m} (W_{ij})^2} * \sqrt{\sum_{j=1}^{m} (W_{lj})^2}} \tag{15}$$

where i, l=1,…n.

Algorithm 4. Similarity matrix.

```
procedure SIMILARITY MATRIX({S})
    for i←1, N do                 ▹ for the all sentences
        for j←i+1, N−1 do         ▹ for the consecutive sentences
            SimMat(i,j) = CosineSimilarity(S_i, S_j)   ▹ Calculate cosine similarity
        end for
    end for
end procedure
```

Similarity matrix can be composed using similarity metrics. The lower triangular values are made zero to avoid backward traversal in the graph and to preserve ordering in sentences while extraction. The diagonal elements are made zero, as there is no self loop.

Please cite this article as: K. Nandhini, S.R. Balasundaram, Extracting easy to understand summary using differential evolution algorithm, Swarm and Evolutionary Computation (2014), http://dx.doi.org/10.1016/j.swevo.2013.12.004

4. Differential evolution

Differential evolution (DE) is a newly developed simple and efficient population-based heuristic and has the great performance of convergence and robustness. As one of the efficient approaches for complicated optimization problems, the proposed differential evolution algorithm is given in the following algorithm

Algorithm 5. Differential evolution.

procedure DE ▹ Basic Differential Evolution
  Create Initial population
  for $i \leftarrow 1$, *size of population* **do**
    for $j \leftarrow 1$, *size of chromosome* **do**
      $\xi_j = S_{r1}, S_{r2} \ldots S_{rn}$ ▹ Individual Chromosome
  end for ▹ Randomly chosen initial combination vector
  end for ▹ Initial population
  repeat
    for $i \leftarrow 1$, *size of population* **do**
      Select X_{ra}, X_{rb}, X_{rc} from the population
      $M_{ra-rc} = (X_{g,ra} \ominus X_{g,rc})$
      Generate Mutant Vector
      $V_{g,i} = X_{g,rb} \oplus F(M_{ra-rc})$
      perform Crossover and Generate trial vector $U_{g,i}$ using $V_{g,i}$ and $X_{g,rb}$
      $U_{g,i}$ is considered for next generation if better than $X_{g,i}$ otherwise $X_{g,i}$ is considered.
    end for
  until Termination Criteria ▹ Max generation is reached
  optimal sentence combination of the chromosome ▹ Candidate Summary
  return *SummarySentences*
end procedure

4.1. Problem formulation

The objective of this work is to maximize the average informative score of the summary that are having better cohesion. The objective function is as follows:

$$Maximize \sum_{i=1}^{n} \sum_{j=i+1}^{n-1} I_{ij} * S_{ij} \tag{16}$$

subject to constraints

- $i > j$
- $S_{ij} < t$ & $S_{ij} \geq T$

constraints ensure only forward traversal in the graph in order to improve readability and to extract cohesive nonredundant candidate sentences. "t" and "T" represent the lower bound and upper bound threshold for setting the similarity of the consecutive sentences.

4.1.1. Chromosome encoding

A chromosome of the proposed DE consists of sequence number of integers that represent the sentence number, which produce the order of sentence in the summary. The gene of the first and last loci is always reserved for the starting sentence (first) and the last sentence. The length of the chromosome is fixed and is decided by compression ratio. A chromosome encodes the problem by listing up the order of sentence numbers from first to last based on its information and cohesion score. The cohesion can be measured using similarity between the consecutive sentences in the list. The similarity matrix can be constructed using the cosine similarity between the sentences.

The gene of the first locus encodes the first sentence and the gene of the second is heuristically selected from the nodes connected with the first node. The sentence is selected based on two constraints that the chronological ordering of the sentences is always preserved by making the lower triangle of the similarity matrix as zero.

4.1.2. Initial population

In order to generate initial population, the size of the population and the procedure to initialize the population play a vital role in generating initial population. Using heuristic initialization yields high mean fitness and finds solution faster. Moreover, the chromosome represents the sentence number and the order of the sentence simultaneously. Therefore heuristic initialization is effected in this paper.

4.1.3. Fitness function

The fitness function must accurately measure the quality of the chromosomes in the population. The fitness function is to find the combination of sentences which produces maximum informative score and cohesion:

$$F_1 = \sum_{i=1}^{n} \frac{1}{\sum_{j=i+1}^{n-1} W_i * F_i} \tag{17}$$

$$F_2 = \sum_{i=1}^{n} \frac{1}{\sum_{j=i+1}^{n-1} Sim(S_i, S_j)} \tag{18}$$

4.1.4. The basic differential evolution algorithm

The mutation operator of DE uses the random samples from the population to perturb the best chromosome of the current population. It adds an amount obtained the difference of two randomly chosen individuals of the current population rather than probability function. The basic DE algorithm is given in Algorithm 5.

4.1.5. Modified mutation

The difference between the random chromosomes (X_{ra} and Xrc) are considered as potential mutation sites and is modified without violating the constraints by preserving the sentence ordering. The algorithm for modified mutation is given in Algorithm 6.

Algorithm 6. Modified mutation.

```
procedure MODIFIED MUTATION                ▹ Modified Mutation
Select two random chromosomes {X_ra, X_rc} from P and the current best chromosome X_rb
  for i ← 2, n−1 do
    M[i] = (Diff(X_ra[i], X_rc[i]))        ▹ Potential Mutation sites
  end for
  for i ← 2, n−1 do
    if (M[i] < > 0) then
      if ((X_rb[i+1]) − (X_rb[i−1]) > 2) then
      Generate R_num
        if ((R_num > (X_rb[i−1])).AND.(R_num < (X_rb[i+1])).AND.(R_num < > (X_rb[i]))) then
          X_rb[i] = R_num
        end if
      end if
    end if
  end for
end procedure
```

4.1.6. Modified crossover

The crossover between trial and current best chromosome can be performed by finding potential crossover sites. The position where the sentence number is equal to or greater than trial chromosome′s number position and the consecutive bit should be greater than current one in order to preserve ordering and to produce potential next generation offspring. The algorithm for performing crossover is given in Algorithm 7.

Algorithm 7. Modified crossover.

```
procedure MODIFIED CROSSOVER               ▹ V_gi ⊗ X_rb
  Find the potential crossing sites
  for i ← 2, n−1 do
    if ((X_rb[i] > = V_g[i]).AND.(X_rb[i] < V_g[i+1])) then ▹
Potential crossing sites
      CS[k] = i
    end if
  end for
Rand(Cr)    ▹ Generate a random number for performing
crossover in random position
  for i ← 1, n do
    U_gi = X_rb[1 : i] ⊕ V_g[i+1 : n]
  end for
end procedure
```

4.1.7. Selection

DE uses the principle of “Survival of the fittest” in its selection process, where if the new chromosome after mutation and crossover is better than current one then it replaces the current best otherwise it continues as best for next generation also.

4.1.8. Termination criteria

The termination criterion of DE could be any one of the following, when there is no change in solution for a certain number of consecutive generations, or a specified or maximum number of iterations. We used number of iterations as termination criteria.

4.2. Experiments and results

For comparison purpose, the same encoded initial population is used for genetic algorithm. The candidates for crossover are chosen using ranking and crossover was accomplished by modified position based crossover. The procedure for position based crossover is as follows: The potential crossover points are identified by checking the identical sentence number in the similar positions of the parent chromosomes X and Y. Selecting randomly from potential points, the offspring was generated by copying the elements from X before the point and from Y after the point for first one and vice versa for the second one. This guarantees that no redundancy and mainly preserves ordering. In case of mutation, random bit is selected and replaced by a sentence number generated randomly that preserves ordering.

4.2.1. Dataset

A collection of hundred articles from educational text of grade four to grade seven are used for evaluation. The texts are mainly from science and social subjects because the magnitude of concepts in social studies and abstract concepts of the science are the challenges faced by the reading difficulties. The text basically belongs to two levels of language one is from ESL text books [47] and another is of graded text [48]. The statistics of the corpus is shown in Table 1.

The summaries of this text are created manually by three independent human annotators. The significance of each sentence to be included in summary is ranked by the annotators for its importance and readability. The sentences that score maximum rank are included in summary according to the compression ratio.

4.3. Evaluation

The summary can be evaluated through direct or intrinsic and indirect or extrinsic [14]. The former are based on direct analysis of the summary by means of some metrics for establishing its quality. Later are based on judgments in terms of how useful the summaries are when specific task is carried out. In Intrinsic evaluation, accuracy, cohesion and readability are considered for comparing the performance of proposed method with Baseline

Table 1
Statistics of the dataset.

Parameter	Size
Number of docs	100
Average no. of sent. per doc.	18
Maximum no. of sent. per doc.	38
Minimum no. of sent. per doc.	11
Summary as % of doc. length	30%
Average summary size (in no. of sent)	13
Maximum no. of sent. per summary	15
Minimum no. of sent. per summary	8

method (Lead- extracts top N sentences according to compression rate) and using Genetic Algorithm. The comparison can be made with an extract of a system with the gold summary generated by considering both important and readability as major criteria for selection.

4.3.1. Accuracy

The accuracy of the baseline method (Lead), GA and the Proposed method at various compression rate (CR) are shown in Fig. 1. As the compression rate increases the average accuracy of all the methods increases. The baseline method has the average accuracy of 0.39, 0.44 and 0.49 at 10%, 20% and 30% compression respectively. When comparing MDE with GA, the relative improvement of MDE over GA is 17.6%, 19.1% and 21.2% respectively. However the performance of GA is equally good in few text. The reason for the DE performance is due to the operation of mutation, which explores all possible combination of sentences that yields the optimal solution. The objective is to extract easy to understand summary can be better evaluated by its cohesion and in turn readability of the summary.

4.3.2. Cohesion

The cohesion of the consecutive sentences helps in understanding the complete text better. Here, the cohesion is measured using cosine similarity. The average similarity of the sentences is considered for measuring the performance of the summary. Here in baseline method, as the compression rate increases the cohesion decreases due to the nature of the text and increase in the number of sentences (Figs. 2). Whereas the selection criteria depend mainly on cosine similarity in both (GA and MDE) the approaches. The reason for GA's lower performance is due its position based crossover that reduces the possibility of exploration. The average cohesion of the proposed method is not reduced due to increase in compression rate when compared to baseline method.

4.3.3. Readability

For predicting the readability of the text the metrics we considered are FOG [43], SMOG [49] that use average sentence length and the percentage of words with at least three syllable as parameters, Flesch-Kincaid [38] uses average sentence length and average syllables per word to calculate the grade level of a text. As the existing metrics does not use all the features we proposed to measure the readability of the text. we measure the readability using the aforementioned metrics (Figs. 3–5). In case of Flesch Kincaid, the proposed approach predicts lower grade in most of the text and lead predicts the remaining. The nature of the text plays a vital role in predicting the readability of the text. SMOG predicts better readability for both the MDE and GA generated summary. FOG index predicts lower grade for MDE generated summary. The performance

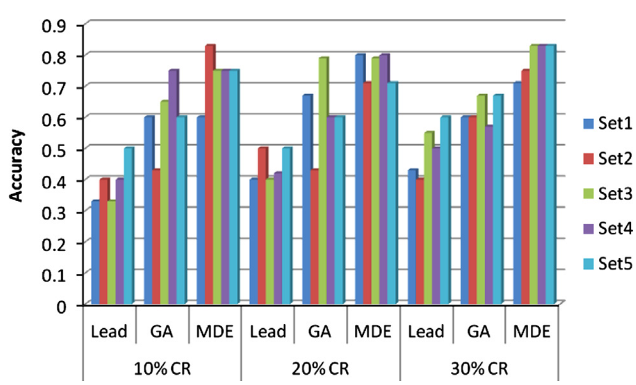

Fig. 1. Comparison of accuracy in various summarization methods.

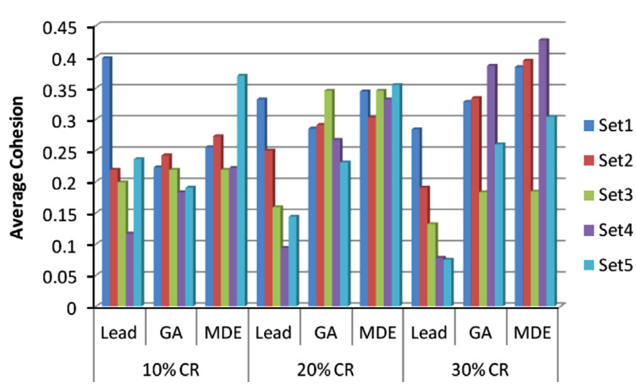

Fig. 2. Comparison of cohesion in various summarization methods.

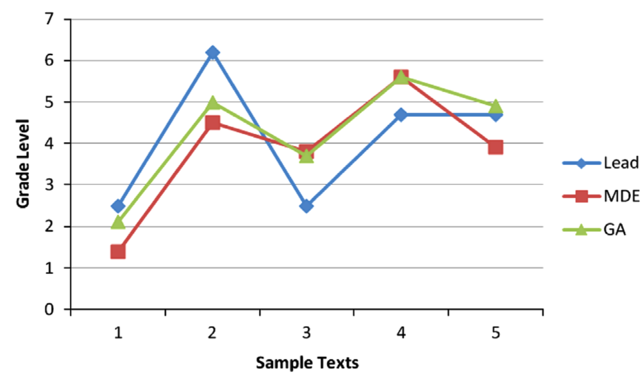

Fig. 3. Comparison of Flesch Kincaid-readability in various summarization methods.

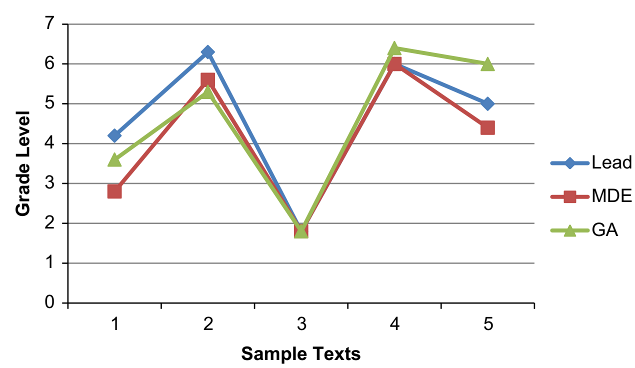

Fig. 4. Comparison of SMOG-readability in various summarization methods.

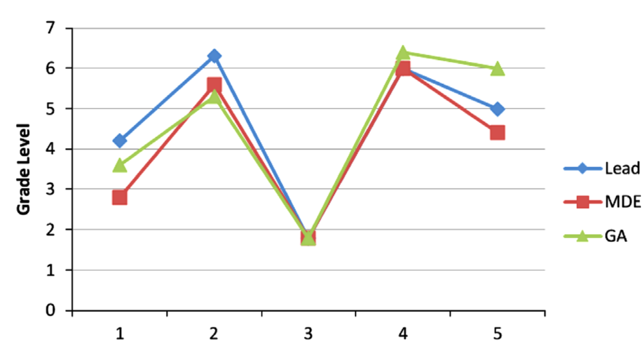

Fig. 5. Comparison of FOG-readability in various summarization methods.

Please cite this article as: K. Nandhini, S.R. Balasundaram, Extracting easy to understand summary using differential evolution algorithm, Swarm and Evolutionary Computation (2014), http://dx.doi.org/10.1016/j.swevo.2013.12.004

Table 2
ANOVA Analysis of the difference between the performance of the students with and without summary(MDE).

Source	*SS*	*df*	*MS*	*F*	*P*
Comprehension test					
Between groups	3.33	1	3.33	7	0.013
Within groups	13.33	28	0.47		

Table 3
ANOVA Analysis of the difference between the performance of the students with two summaries(GA and MDE).

Source	*SS*	*df*	*MS*	*F*	*P*
Comprehension test					
Between groups	2.13	1	2.13	6.13	0.019
Within groups	9.73	28	0.34		

of the MDE is better than GA and Baseline method in all the evaluations (accuracy, readability and cohesion). The reason is due to the nature of mutation operation in MDE. When comparing MDE with GA, the convergence of MDE is very slower than GA. In spite of its slow convergence, it is robust and will continue to improve until an optimal solution is found. Due to the overhead caused by the computational complexity of DE, the time it takes to complete a single generation is larger than GA. The sentence ordering problem still increases the complexity of DE. The reason for better performance is that is each permutation goes through a much involved type of mutation than mutation in GA. The major reason for GA's reduced performance is due to its position based crossover strategy which reduces the probability of the crossover in many sites. As the summary is focused on reading difficulties, it would be better if task based evaluation is performed using the target audience.

4.3.4. *Task based evaluation*

To evaluate the satisfaction of the learners about the summary generated from various methods for overviewing the text. A way to measure a text's difficulty level is to ask the learner various subjective questions in which they must evaluate how easy it is to understand a text. This meta-comprehension of text has been measured using Likert scales in educational studies [50]. User evaluation using likert scale point is an indirect way of measuring text readability. So, text readability can be evaluated directly to measure the comprehension of information from a text: objective comprehension questions, such as multiple-choice or yes/no 5 questions, both of which have been used with learners in educational and psychological studies of comprehension [15].

We carried out the experiments with 30 learners with reading difficulties range from fourth grade to seventh grade. Each learner is given two texts one from text books with comprehension questions and another with summary followed by text and objective questions from grade level text. The effect of summary assists reading on students with reading difficulties in enhancing readability and comprehension for both types of summaries are given in Table 2. From the result, it is clear that the effect of summary in answering questions is significant than without summary text. The effect of two approaches is also studied and shows significant difference due its improved cohesiveness (Table 3). From the results it is evident that the effect of summary has positive effect in enhancing readability in turn comprehension of a given text. The learner is given easy to read important sentences which increases interest and motivation. The objective questions focus much on evaluating recall, recognize questions and on analysis or comprehension level of blooms taxonomy. it is more clear from the questions answered by the learners that summary helps them in answering better in terms of factual questions than in analysis questions. Moreover the summary that is easy to one learner may not be same for all. In future, we plan to develop an automatic summarization with various levels that can be better suited to a specific range of learners.

5. Conclusion

Researches show that the effect of summarization strategy in comprehending a given text for reading difficulties is significant. An automatic summarizer helps the reading difficulties in understanding the text better. In order to improve the readability of the given text, easy to understand summary is extracted using proposed algorithm.

The proposed algorithm considers informativeness, readability and cohesion for extracting easy to understand summary using Differential Evolution. The summary generated using proposed algorithm performs better than baseline and corpus based approach. When the target audience is specific, the usability of summarization can be measured through task based evaluation. The task based evaluation shows significant effect of summary in improving readability of the target audience. Furthermore the summary can be used as overview before reading the complete text which in turn improves readability and comprehension. when the text is complex, the summary generated from the text is also complex. It is better that summary can be simplified using simplification technique to help the audience in a better way. Our future work is to develop individualized summary using interactive differential evolution algorithm with simplification technique.

References

[1] U. Hahn, I. Mani, The challenges of automatic summarization, Computer 33 (11) (2000) 29–36.
[2] J. Kupiec, J. Pedersen, F. Chen, A trainable document summarizer, in: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1995, pp. 68–73.
[3] J. Yeh, H. Ke, W. Yang, I. Meng, et al., Text summarization using a trainable summarizer and latent semantic analysis, Inf. Process. Manage. 41 (1) (2005) 75–95.
[4] R. Mihalcea, H. Ceylan, Explorations in automatic book summarization, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 380–389.
[5] R. Aliguliyev, Automatic document summarization by sentence extraction, Comput. Technol. 12 (5).
[6] H. Luhn, The automatic creation of literature abstracts, IBM J. Res. Dev. 2 (2) (1958) 159–165.
[7] P. Baxendale, Machine-made index for technical literature an experiment, IBM J. Res. Dev. 2 (4) (1958) 354–361.

Please cite this article as: K. Nandhini, S.R. Balasundaram, Extracting easy to understand summary using differential evolution algorithm, Swarm and Evolutionary Computation (2014), http://dx.doi.org/10.1016/j.swevo.2013.12.004

[8] H. Edmundson, New methods in automatic extracting, J. ACM (JACM) 16 (2) (1969) 264–285.

[9] C. Aone, M. Okurowski, J. Gorlinsky, Trainable, scalable summarization using robust NLP and machine learning, in: Proceedings of the 17th International Conference on Computational Linguistics, Association for Computational Linguistics, vol. 1, 1998, pp. 62–66.

[10] C. Lin, E. Hovy, Identifying topics by position, in: Proceedings of the Fifth Conference on Applied Natural Language Processing, Association for Computational Linguistics, 1997, pp. 283–290.

[11] D. Reutzel, J. Smith, P. Fawson, An evaluation of two approaches for teaching reading comprehension strategies in the primary years using science information texts, Early Child. Res. Quart. 20 (3) (2005) 276–305.

[12] G. Grefenstette, Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind, in: Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization, 1998, pp. 111–118.

[13] V. Vandeghinste, Y. Pan, Sentence compression for automated subtitling: a hybrid approach, in: Proceedings of the ACL Workshop on Text Summarization, 2004, pp. 89–95.

[14] K. Spärck Jones, Automatic summarising: the state of the art, Inf. Process. Manage. 43 (6) (2007) 1449–1481.

[15] A.W. Graves, Effects of direct instruction and metacomprehension training on finding main ideas, Learn. Disabil. Res. 1 (2) (1986) 90–100.

[16] M. Gajria, J. Salvia, The effects of summarization instruction on text comprehension of students with learning disabilities, Except. Child. 58 (6) (1992) 508–516.

[17] M. Fattah, F. Ren, GA, MR, FFNN, PNN and GMM based models for automatic text summarization, Comput. Speech Lang. 23 (1) (2009) 126–144.

[18] Y. Gong, X. Liu, Generic text summarization using relevance measure and latent semantic analysis, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2001, pp. 19–25.

[19] D. Radev, E. Hovy, K. McKeown, Introduction to the special issue on summarization, Comput. Linguist. 28 (4) (2002) 399–408.

[20] R. Mihalcea, P. Tarau, A language independent algorithm for single and multiple document summarization, in: Proceedings of IJCNLP, vol. 5, 2005.

[21] G. Salton, A. Singhal, M. Mitra, C. Buckley, Automatic text structuring and summarization, Inf. Process. Manage. 33 (2) (1997) 193–207.

[22] I. Mani, E. Bloedorn, Summarizing similarities and differences among related documents, Inf. Retr. 1 (1–2) (1999) 35–67.

[23] R. Mihalcea, Language independent extractive summarization, in: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, 2005, pp. 49–52.

[24] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web.

[25] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM (JACM) 46 (5) (1999) 604–632.

[26] E. Filatova, V. Hatzivassiloglou, A formal model for information selection in multi-sentence text extraction, in: Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics, 2004, p. 397.

[27] S. Kirkpatrick, D.G. Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[28] F. Glover, Tabu search – part i, ORSA J. Comput. 1 (3) (1989) 190–206.

[29] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.

[30] M. Dorigo, G. Di Caro, Ant colony optimization: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99, vol. 2, IEEE, 1999.

[31] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.

[32] G. Onwubolu, D. Davendra, Scheduling flow shops using differential evolution algorithm, Eur. J. Oper. Res. 171 (2) (2006) 674–692.

[33] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.

[34] S. Islam, S. Das, S. Ghosh, S. Roy, P.N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Trans. Cybern. Syst. Man Cybern. Part B 42 (2) (2012) 482–500.

[35] J.M. Conroy, J.D. Schlesinger, D.P. O′leary, J. Goldstein, Back to basics: classy 2006, in: Proceedings of DUC, vol. 6, 2006.

[36] G. Onwubolu, D. Davendra, Differential Evolution for Permutation—Based Combinatorial Problems, Springer, 2009.

[37] G. Erkan, D. Radev, Lexrank: graph-based lexical centrality as salience in text summarization, J. Artif. Intell. Res. (JAIR) 22 (2004) 457–479.

[38] J. Kincaid, R. Fishburne, R.L., Chissom, B.S. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel, Research Branch Report 8-75, 1975.

[39] R. Gunning, The Technique of Clear Writing, McGraw-Hill, New York, 1952.

[40] L. Feng, N. Elhadad, M. Huenerfauth, Cognitively motivated features for readability assessment, in: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2009, pp. 229–237.

[41] R. Davis, E. Braun, The Gift of Dyslexia: Why Some of the Smartest People Can′t Read... and How They Can Learn, Penguin Group, 2010.

[42] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manage. 24 (5) (1988) 513–523.

[43] R. Gunning, The fog index after twenty years, J. Bus. Commun. 6 (2) (1969) 3–13.

[44] A. Kano, Evaluating phrase sets for use with text entry method evaluation with dyslexic participants, in: Workshop on Improving and Assessing Pen-Based Input Techniques at British HCI, 2005.

[45] B. Jann, Making regression tables from stored estimates, J. Stata 5 (3) (2005) 288.

[46] W. Richard, Review of regression models for categorical dependent variables using Stata, Second Edition, by Long and Freese, J. Stata 6 (2) (2006) 273–278.

[47] URL ⟨http://www.samacheerkalvi.in⟩.

[48] URL ⟨http://www.impact-information.com/scales.pdf⟩.

[49] G. Mc Laughlin, Smog grading-a new readability formula, J. Read. (1969) 639–646.

[50] R. Maki, S. Berry, et al., Metacomprehension of text material, J. Exp. Psychol.: Learn. Mem. Cogn. 10 (4) (1984) 663–679.