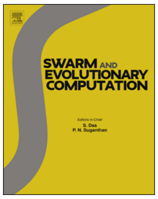
Swarm and Evolutionary Computation 16 (2014) 19–27

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
Available online 21 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 B.V. 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.

```
procedure SUMMARY EXTRACTION(.txt file)
    Preprocessing ▹ Sentence Segmentation, Tokenizer, Parts of speech tagging, Stemming
    Feature Extraction ▹ Readability,Informative and Cohesive Features
    Representing Data in VSM ▹ Vector Space Model
    Calculating Informative Score ▹ Sum of Weighted Score of Features
    Applying Differential Evolution algorithm ▹ Target Population
    Finding optimal sentence combination ▹ Combinatorial Optimization
    return Candidate Sentences ▹ Summary sentences
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.