Intelligent Information Management, 2009, 1, 128-138
doi:10.4236/iim.2009.12019 Published Online November 2009 (http://www.scirp.org/journal/iim)

Scientific Research

Evolutionary Algorithm for Extractive Text Summarization

Rasim ALGULIEV, Ramiz ALIGULIYEV
Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan
Email: *rasim@science.az, a.ramiz@science.az*

Abstract: Text summarization is the process of automatically creating a compressed version of a given document preserving its information content. There are two types of summarization: extractive and abstractive. Extractive summarization methods simplify the problem of summarization into the problem of selecting a representative subset of the sentences in the original documents. Abstractive summarization may compose novel sentences, unseen in the original sources. In our study we focus on sentence based extractive document summarization. The extractive summarization systems are typically based on techniques for sentence extraction and aim to cover the set of sentences that are most important for the overall understanding of a given document. In this paper, we propose unsupervised document summarization method that creates the summary by clustering and extracting sentences from the original document. For this purpose new criterion functions for sentence clustering have been proposed. Similarity measures play an increasingly important role in document clustering. Here we've also developed a discrete differential evolution algorithm to optimize the criterion functions. The experimental results show that our suggested approach can improve the performance compared to sate-of-the-art summarization approaches.

Keywords: sentence clustering, document summarization, discrete differential evolution algorithm

1. Introduction

Text summarization is the process of automatically creating a compressed version of a given document preserving its information content. Automatic document summarization is an important research area in natural language processing (NLP). The technology of automatic document summarization is developing and may provide a solution to the information overload problem [1–3].

The process of text summarization can be decomposed into three phases: analysis, transformation, and synthesis. The analysis phase analyzes the input text and selects a few salient features. The transformation phase transforms the results of the analysis into a summary representation. Finally, the synthesis phase takes the summary representation, and produces an appropriate summary corresponding to users' needs. In the overall process, compression rate, which is defined as the ratio between the length of the summary and that of the original, is an important factor that influences the quality of the summary. As the compression rate decreases, the summary will be more concise; however, more information is lost. While the compression rate increases, the summary will be larger; relatively, more insignificant information is contained. In fact, when the compression rate is 5–30%, the quality of the summary is acceptable [1–4].

Text summarization can be categorized into two approaches: extractive and abstractive. Extractive summarization methods simplify the problem of summarization into the problem of selecting a representative subset of the sentences in the original documents. Abstractive summarization may compose novel sentences, unseen in the original sources [1]. However, abstractive approaches require deep NLP such as semantic representation, inference and natural language generation, which have yet to reach a mature stage nowadays [5].

Extractive summarization systems are commonly used in automatic summarization to produce extractive summaries. Systems for extractive summarization are typically based on technique for sentence extraction, and attempt to identify the set of sentences that are most important for the overall understanding of a given document. Most commonly, such systems use some kind of similarity or centrality metric to identify the set of sentences to include in the summary [1,4,6–15]. For example, in [14] a paragraph extraction from a document based on intra-document links between paragraphs is proposed. It yields a TRM (Text Relationship Map) from intra-links, which indicate that the linked texts are semantically related. It proposes four strategies from the TRM: bushy path, depth-first path, segmented bushy path, augmented segmented bushy path. An improved version of this approach is proposed in [1,4,6].

In this paper we demonstrate an extractive text sum-

marization method which is based on sentence clustering. For this purpose new criterion functions for sentence clustering have been offered. In our study we developed a discrete differential evolution algorithm to optimize the criterion functions. The experimental results on an open benchmark datasets from DUC2001 and DUC2002 show that our suggested approach can improve the performance compared to state-of-the-art summarization approaches.

The rest of this paper is organized as follows: Section 2 introduces related works. The proposed sentence clustering based approach for generic single-document summarization is presented in Section 3. The discrete differential evolution algorithm for optimization procedure is given in Section 4. The experiments and results are given in Section 5. Finally, we conclude our paper in Section 6.

2. Related Work

Automatic document summarization has been actively investigated in recent years, and most researchers have concentrated on the extractive summarization method, but not the abstractive summarization method – see for example, the 6th issue of the Information Processing and Management: an International Journal of 2007. The current paper contains four references to the articles published in this edition [5,16–18]. A comprehensive survey of document summarization can be found in [17].

The centroid-based method [19,13] is one of the most popular extractive summarization methods. MEAD (http://www.summarization.com/mead/) is an implementation of the centroid-based method for either single- or multi-document summarizing. It is based on sentence extraction. For each sentence in a cluster of related documents, MEAD computes three features and uses a linear combination of the three to determine what sentences are most salient. The three features used are centroid score, position, and overlap with first sentence (which may happen to be the title of a document). For single documents or (given) clusters it computes centroid topic characterizations using tf-idf-type data. It ranks candidate summary sentences by combining sentence scores against centroid, text position value, and tf-idf title/lead overlap. Sentence selection is constrained by a summary length threshold, and redundant new sentences avoided by checking cosine similarity against prior ones [20]. In [21] each document is considered as a sequence of sentences and the objective of extractive summarization is to label the sentences in the sequence with 1 and 0, where a label of 1 indicates that a sentence is a summary sentence while 0 denotes a non-summary sentence. To accomplish this task, a conditional random field is applied [22]. A novel extractive approach based on manifold-ranking of sentences to query-based multi-document summarization proposed in [23]. This approach first uses the manifold-ranking process to compute the manifold-ranking score for each sentence that denotes the biased information richness of the sentence, and then uses greedy algorithm to penalize the sentences with highest overall scores, which are considered both informative and novel, and highly biased to the given query.

The summarization techniques can be classified into two groups: supervised techniques, that rely on machine learning algorithms trained on pre-existing document-summary pairs, and unsupervised techniques, based on properties and heuristics derived from the text. Supervised extractive summarization techniques [1,4–6,21, 24–26] treat the summarization task as a two-class classification problem at the sentence level, where the summary sentences are positive samples while the non-summary sentences are negative samples. After representing each sentence by a vector of features, the classification function can be trained in two different manners [21]. The first is in a discriminative way with well-known algorithms such as SVM (Support Vector Machine) [4]. In [1], the use of genetic algorithm (GA), mathematical regression (MR), feed forward neural network (FFNN), probabilistic neural network (PNN) and Gaussian mixture model (GMM) for automatic text summarization task have been investigated. This approach is a trainable summarizer, which takes into account several features, including sentence position, positive keyword, negative keyword, sentence centrality, sentence resemblance to the title, sentence inclusion of name entity, sentence inclusion of numerical data, sentence relative length, bushy path of the sentence and aggregated similarity for each sentence to generate summaries. The article [25] presents a multi-document, multilingual, theme-based summarization system based on modeling text cohesion (story flow). Many unsupervised methods have been developed for document summarization by exploiting different features and relationships of the sentences, such as clustering of sentences [7–11], the hidden topics in the documents [12], graphs based on the similarity of sentences[13,19,27].

Recently, graph-based methods have been offered to rank sentences. Lexrank [19] and [27] are two such systems using the algorithms PageRank and HITS to compute sentence importance. Lexrank is used to compute sentence importance based on the concept of eigenvector centrality in a graph representation of sentences for multi-document summarization task. The graph-based extractive summarization algorithms succeed in identifying the most important sentences in a text based on information exclusively drawn from the text itself. Unlike other systems, which attempt to find out what makes a good summary by training on collections of summaries built for other articles, the graph-based methods are fully unsupervised, and rely on the given texts to derive an extractive summary[23].

On the other hand, summarization task can also be categorized as either generic or query-based. A query-based summary presents the information that is most relevant to the given queries [16,23,24,28,29] while a generic summary gives an overall sense of the document's content [7–14,19,27,30]. The QCS system (Query, Cluster, and Summarize)[16] performs the following tasks in response to a query: retrieves relevant documents; separates the retrieved documents into clusters by topic, and creates a summary for each cluster. QCS is a tool for document retrieval that presents results in a format so that a user can quickly identify a set of documents of interest. In [29] are developed a generic, a query-based, and a hybrid summarizer, each with differing amounts of document context. The generic summarizer used a blend of discourse information and information obtained through traditional surface-level analysis. The query-based summarizer used only query-term information, and the hybrid summarizer used some discourse information along with query-term information.

Automatic document summarization is a highly interdisciplinary research area related with computer science, multimedia, statistics, as well as cognitive psychology. In [31] is introduced an intelligent system, the event indexing and summarization (EIS) system, for automatic document summarization, which is based on a cognitive psychology model (the event-indexing model) and the roles and importance of sentences and their syntax in document understanding. The EIS system involves syntactic analysis of sentences, clustering and indexing sentences with five indices from the event-indexing model, and extracting the most prominent content by lexical analysis at phrase and clause levels.

3. Sentence Clustering

Clustering is the process of discovering natural groupings or clusters and identifying interesting distributions and patterns within multidimensional data based on some similarity measure. The topic of clustering has been extensively studied in many scientific disciplines such as text mining, pattern recognition, IR etc. Document clustering is a central problem in text mining which can be defined as grouping documents into clusters according to their topics or main contents. Document clustering has many purposes including expanding a search space, generating a summary, automatic topic extraction, browsing document collections, organizing information in digital libraries and detecting topics. In the literature a wide variety of clustering algorithms have been proposed for different applications and sizes of data sets. The surveys on the topics [32–35] offer a comprehensive summary of the different applications and algorithms.

Generally clustering problems are determined by four basic components [36,37]: 1) the (physical) representation of the given data set; 2) the distance/dissimilarity measures between data points; 3) the criterion/objective function which the clustering solutions should aim to optimize; and, 4) the optimization procedure. For a given data clustering problem, the four components are tightly coupled. Various methods/criteria have been proposed over the years from various perspectives and with various focuses.

3.1. Sentence Similarity Measure Based on Terms Co-Occurrence

Let a document D is decomposed into a set of sentences $D=\{S_1,S_2,...,S_n\}$, where n is the number of sentences. Let $T=\{t_1,t_2,...,t_m\}$ represents all the distinct words (terms) occurring in a document D, where m is the number of words. In most existing document clustering algorithms, documents are represented using the vector space model (VSM) [33]. Each document is represented using these words as a vector in m-dimensional space. A major characteristic of this representation is the high dimensionality of the feature space, which imposes a big challenge to the performance of clustering algorithms. They could not work efficiently in high-dimensional feature spaces due to the inherent sparseness of the data [17]. The vector dimension *m* is very large compared to the number of words in a sentence, thus the resulting vectors would have many null components [38]. In our method, a sentence S_i is represented as a set of distinct terms appearing in it, $S_i=\{t_1,t_2,...,t_{m_i}\}$, where m_i is the number of distinct terms in the sentence S_i.

Similarity measures play an increasingly important role in NLP and IR. Similarity measures have been used in text-related research and application such as text mining, information retrieving, text summarization, and text clustering. These applications show that the computation of sentence similarity has become a generic component for the research community involved in knowledge representation and discovery. There are more papers on similarity between documents than between sentences or short texts, but not few [38,39]. The paper [39] presents a method for measuring the similarity between sentences or very short texts, based on semantic and word order information. First, semantic similarity is derived from a lexical knowledge base and a corpus. Second, the proposed method considers the impact of word order on the sentence meaning. The overall sentence similarity is defined as a combination of semantic similarity and word order similarity. Liu *et al*. [40] present a novel method to measure similarity between sentences by analyzing parts of speech and using Dynamic Time Warping technique. In [41] proposed a novel measure based on the earth

mover's distance (EMD) to evaluate document similarity by allowing many-to-many matching between subtopics. First, each document is decomposed into a set of subtopics, and then the EMD is employed to evaluate the similarity between two sets of subtopics for two documents by solving the transportation problem. The proposed measure is an improvement of the previous optimal matching (OM)-based measure, which allows only one-to-one matching between subtopics. The study of semantic similarity between words has long been an integral part of IR and NLP [42]. The method, proposed in [42], integrates both page counts and snippets to measure semantic similarity between a given pair of words. In this paper modified four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice and PMI (Point-wise Mutual Information), to compute semantic similarity using page counts.

In this section we present a method to measure similarity between sentences using the Normalized Google Distance (NGD) [43]. First we calculate a similarity measure between the terms before defining the similarity measure between the sentences. Using the NGD [43] the similarity measure between terms t_k and t_l we define as:

$$sim_{\mathrm{NGD}}(t_k,t_l)=\exp(-\mathrm{NGD}(t_k,t_l)) \tag{1}$$

where

$$\mathrm{NGD}(t_k,t_l)=\frac{\max\{\log(f_k),\log(f_l)\}-\log(f_{kl})}{\log n-\min\{\log(f_k),\log(f_l)\}}, \tag{2}$$

f_k is the number of sentences containing the term t_k, f_{kl} denotes the number of sentences containing both terms t_k and t_l, n is the number of sentences in the document.

From the properties of NGD follows that [43]:

1) The range of the $diss_{\mathrm{NGD}}(t_k,t_l)$ is between 0 and 1;

- If $t_k=t_l$ or if $t_k\neq t_l$ but $f_k=f_l=f_{kl}>0$, then $sim_{\mathrm{NGD}}(t_k,t_l)=1$. That is, the semantics of t_k and t_l, in the Google sense is the same.
- If $f_k>0$, $f_l>0$ and $f_{kl}=0$, we take $\mathrm{NGD}(t_k,t_l)=1$, then $0<sim_{\mathrm{NGD}}(t_k,t_l)<1$.
- If $0<f_{kl}<f_l<f_k<n$ and $f_k\cdot f_l>n\cdot f_{kl}$, then $0<sim_{\mathrm{NGD}}(t_k,t_l)<1$.

2) $sim_{\mathrm{NGD}}(t_k,t_k)=1$ for any t_k. For every pair t_k and t_l, we have $sim_{\mathrm{NGD}}(t_k,t_l)=sim_{\mathrm{NGD}}(t_l,t_k)$: It is symmetric.

Using the formula (1) we define a similarity measure between sentences S_i and S_j as follows:

$$sim_{\mathrm{NGD}}(S_i,S_j)=\frac{\sum_{t_k\in S_i}\sum_{t_l\in S_j}sim_{\mathrm{NGD}}(t_k,t_l)}{m_i m_j} \tag{3}$$

From the properties of $sim_{\mathrm{NGD}}(t_k,t_l)$ follows that: 1) the range of the $sim_{\mathrm{NGD}}(S_i,S_j)$ is in 0 and 1; 2) $sim_{\mathrm{NGD}}(S_i,S_i)\geq 0$ for every S_i; 3) for every pair S_i and S_j $sim_{\mathrm{NGD}}(S_i,S_j)=sim_{\mathrm{NGD}}(S_j,S_i)$: it is exchangeable.

3.2. Objective Functions

Typically clustering algorithms can be categorized as agglomerative or partitional based on the underlying methodology of the algorithm, or as hierarchical or flat (non-hierarchical) based on the structure of the final solution [32–35]. A key characteristic of many partitional clustering algorithms is that they use a global criterion function whose optimization drives the entire clustering process. In recent years, it has been recognized that the partitional clustering technique is well suited for clustering a large document database due to their relatively low computational requirements [44].

Automatic clustering is a process of dividing a set of objects into unknown groups, where the best number k of groups (or clusters) is determined by the clustering algorithm. That is, objects within each group should be highly similar to each other than to objects in any other group. The automatic clustering problem can be defined as follows [32–35,45]:

The set of sentences $D=\{S_1,S_2,...,S_n\}$ are clustered into non-overlapping groups $C=\{C_1,...,C_k\}$, where C_k is called a cluster, k is the unknown number of clusters. The partition should possess three properties:

1) Two different clusters should have no sentences in common, i.e. $C_p\cap C_q=\varnothing$ for $\forall p\neq q \quad p,q\in\{1,2,...,k\}$;

2) Each sentence should definitely be attached to a cluster, i.e. $\bigcup_{p=1}^{k}C_p=D$;

3) Each cluster should have at least one sentence assigned, i.e. $C_p\neq\varnothing \quad \forall p\in\{1,2,...,k\}$.

Partitional clustering can be viewed as an optimization procedure that tries to create high-quality clusters according to a particular criterion function. Criterion functions used in partitional clustering reflect the underlying definition of the "goodness" of clusters. Many criterion functions have been proposed in the literature [32–35,44] to produce more balanced partitions.

We introduce a criterion function that is defined as follows:

$$F=(1+sigm(F_1))^{F_2}\rightarrow\max \tag{4}$$

where $sigm(z)$ is a sigmoid function that maps from the real numbers into $[0,1]$, $sigm(z)=\frac{1}{1+\exp(-z)}$.

The criterion function (4) balances both intra-cluster similarity and inter-cluster dissimilarity. This function is obtained by combining two criteria:

$$F_1=\sum_{p=1}^{k}\left|C_p\right|\sum_{S_i,S_l\in C_p}sim_{\mathrm{NGD}}(S_i,S_l)\to\max \qquad (5)$$

and

$$F_2=\sum_{p=1}^{k-1}\frac{1}{\left|C_p\right|}\sum_{q=p+1}^{k}\frac{1}{\left|C_q\right|}\sum_{S_i\in C_p}\sum_{S_l\in C_q}sim_{\mathrm{NGD}}(S_i,S_l)\to\min \qquad (6)$$

The F_1 criterion Function (5) maximizes the average sum of the pairwise similarity between the sentences assigned to each cluster. The F_2 criterion Function (6) computes the clustering by finding a solution that separates each cluster from other clusters. It minimizes the similarity between the sentences S_i and S_l assigned to different clusters C_p and C_q, respectively.

4. Modified Discrete Differential Evolution Algorithm (MDDE)

There are many techniques that can be used to optimize the criterion Functions (4)-(6) described in the previous Section 3. The paper [17] presents an up-to-date survey on evolutionary algorithms for clustering. It tries to reflect the profile of this area by focusing more on those subjects that have been given more importance in the literature. Particularly, the paper has focused mainly on hard partitional algorithms, though overlapping (soft/ fuzzy) approaches have also been covered. An original contribution of the present paper is that it discusses key issues on the design of evolutionary algorithms for data partitioning problems, such as usually adopted representations, evolutionary operators, and fitness functions, just to mention a few. In particular, mutation and crossover operators commonly described in the literature are conceptually analyzed, giving especial emphasis to those genetic operators specifically designed for clustering problems (i.e., cluster-oriented and context-sensitive operators). In our study these criterion functions were optimized using a differential evolution (DE) [45,46]. The execution of the differential evolution is similar to other evolutionary algorithms like genetic algorithms or evolution strategies. The evolutionary algorithms differ mainly in the representation of parameters (usually binary strings are used for genetic algorithms while parameters are real-valued for evolution strategies and differential evolution) and in the evolutionary operators.

Like to other evolutionary algorithm, DE also starts with a population of N n-dimensional search variable vectors. The classical DE [45,46] is a population-based global optimization that uses a real-coded representation. In our study we use a genetic encoding that deals with discrete variables (clusters), such that each component of the chromosome takes a value between 1 and k and represents the cluster to which the sentence is assigned. Potential set of solutions to the optimization problem are represented by a population of chromosomes. The initial population of chromosomes is generated by producing the series of integer random numbers. These numbers are uniformly generated between 1 and k inclusively. Potential solutions (chromosomes) to the target problem are encoded as fixed length discrete strings, i.e., $X_r(t)=[x_{r,1}(t),x_{r,2}(t),...,x_{r,n}(t)]$ where $x_{r,s}(t)\in\{1,2,...,k\}$, $r=1,2,...,N$, and $s=1,2,...,n$, N is the size of the population. For example, given the number of clusters $k=4$, the number of sentences $n=8$ and the population size $N=3$, a population can be as: $X_1(t)=[2,1,3,4,2,1,4,3]$, $X_2(t)=[4,3,4,2,2,3,1,1]$, and $X_3(t)=[1,2,4,2,3,1,3,4]$. In this example, chromosome $X_1(t)$ represents a candidate solution where sentences S_2 and S_6 are assigned to cluster C_1; sentences S_1 and S_5 are located in cluster C_2; sentences S_3 and S_8 are assigned to cluster C_3, and S_4 and S_7 are located in cluster C_4.

For each individual vector $X_r(t)$ that belongs to the current population, DE randomly samples three other individuals $X_{r1}(t)$, $X_{r2}(t)$, and $X_{r3}(t)$ from the same generation (for mutually different $r\neq r1\neq r2\neq r3$). It then calculates the difference of $X_{r2}(t)$ and $X_{r3}(t)$, scales it by a scalar λ, and creates a trial offspring $Y_r(t+1)=[y_{r,1}(t+1),y_{r,2}(t+1),...,y_{r,n}(t+1)]$ by adding the result to $X_{r1}(t)$. Thus, for the sth component of each vector

$$y_{r,s}(t+1)=\begin{cases}x_{r1,s}(t)+\lambda(x_{r2,s}(t)-x_{r2,s}(t)), \text{ if } \mathrm{rnd}_s<\mathrm{CR}\\ x_{r,s}(t), \text{ otherwise}\end{cases} \qquad (7)$$

The scaling factor (λ) and the crossover rate (CR) are control parameters of DE, are set by the user. Both values remain constant during the search process. λ is a real-valued factor (usually in range $[0,1]$), that controls the amplification of differential variations and CR is a real-valued crossover factor in range $[0,1]$ controlling the probability to choose mutated value for x instead of its current value. rnd_s is the uniformly distributed random numbers within the range $[0,1]$ chosen

once for each $s \in \{1,2,...,n\}$.

If the new offspring yields a better value of the objective function, it replaces its parent in the next generation; otherwise, the parent is retained in the population, i.e.,

$$X_r(t+1) = \begin{cases} Y_r(t+1), \text{ if } f(Y_r(t+1)) > f(X_r(t)) \\ X_r(t), \text{ if } f(Y_r(t+1)) \le f(X_r(t)) \end{cases} \quad (8)$$

where $f(\cdot)$ is the objective function to be maximized.

After initialization of the population the process of calculation of the fitness is performed. To judge the quality of a partition provided by a chromosome, it is necessary to have a fitness functions. The fitness functions are defined as

$$fitness_1(X_a(t)) = F_1(X_a(t)) \quad (9)$$

$$fitness_2(X_a(t)) = \frac{1}{F_2(X_a(t))}, \quad (10)$$

$$fitness(X_a(t)) = F(X_a(t)). \quad (11)$$

Maximization of the fitness Functions (9)-(11) leads to maximization (or minimization) of the objective Functions (4)-(6), respectively.

The main difference between the traditional discrete DE (DDE) algorithms and the algorithm proposed here is that it uses the mutation operation adopted from genetic algorithm. The algorithm proposed is based on the mutation adopted from genetic algorithms. In our modification, at the iteration $t+1$ for each vector $X_r(t)$ creates a vector $m_r(t+1) = [m_{r,1}(t+1), m_{r,2}(t+1), ..., m_{r,n}(t+1)]$, which is defined as:

$$m_{r,s}(t+1) = \begin{cases} 1, \text{ if } \text{rand}_s < sigm(y_{r,s}(t+1)) \\ 0, \text{ otherwise} \end{cases} \quad (12)$$

The vector $m_r(t+1)$ represents the changes that will be needed to move the particle from $X_r(t)$ to $X_r(t+1)$. If the component of vector $m_r(t+1)$ is one, it means that this component will be copied from the $X_r(t)$ to $X_r(t+1)$. If the component of $m_r(t+1)$ is null, it means that this component will be mutated. The inversion operator has been used as a mutation operator. The inversion operator takes the components from the $X_r(t)$ corresponding to the null components of the vector $m_r(t+1)$ and puts them to form a new particle. The pseudo-code of an inversion operator is described in Figure 1 ($|S|$ is the cardinality of the set S) [11]:

Let's consider a following example (Figure 2):

```
procedure inversion operator
S = {∅}
for s = 1 : n
    if m_{r,s}(t+1) = 1 then x_{r,s}(t+1) = x_{r,s}(t)
      else
          S = S ∪ {s}
    end_if
end_for
    if |S| > 0 then
        s^+ = max S
        s^- = min S
        x_{r,s^-}(t+1) = x_{r,s^+}(t)
        x_{r,s^+}(t+1) = x_{r,s^-}(t)
        S = S \ {s^+, s^-}
      else if |S| = 0 then stop
    end_if
```

Figure 1. Pseudo-code of an inversion operator

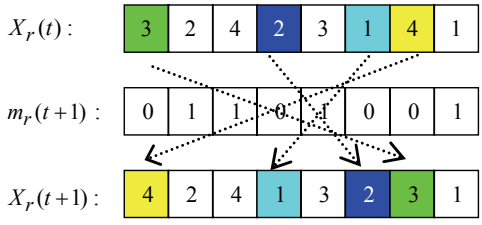

Figure 2. Inversion operator

The components of a vector $X_r(t+1)$, using the pseudo-code described in Figure 1, are calculated as follows:

Step 1. if $m_{r,s}(t+1)=1$ then $x_{r,s}(t+1)=x_{r,s}(t)$, i.e. $x_{r,2}(t+1)=x_{r,2}(t)=2$, $x_{r,3}(t+1)=x_{r,3}(t)=4$, $x_{r,5}(t+1)=x_{r,5}(t)=3$, and $x_{r,8}(t+1)=x_{r,8}(t)=1$

Step 2. $\mathrm{S}=\{s:m_{r,s}(t+1)=0\}=\{1,4,6,7\}$

Step 3. $s^+=\max \mathrm{S}=\max\{1,4,6,7\}=7$, $s^-=\min \mathrm{S}=\min\{1,4,6,7\}=1$

Step 4. $x_{r,s^-}(t+1)=x_{r,s^+}(t)$ and $x_{r,s^+}(t+1)=x_{r,s^-}(t)$, i.e. $x_{r,1}(t+1)=x_{r,7}(t)=4$ and $x_{r,7}(t+1)=x_{r,1}(t)=3$

Step 5. $\mathrm{S}=\mathrm{S}\setminus\{s^+,s^-\}=\{1,4,6,7\}\setminus\{1,7\}=\{4,6\}$

Step 6. $s^+=\max \mathrm{S}=\max\{4,6\}=6$, $s^-=\min \mathrm{S}=\min\{4,6\}=4$

Step 7. $x_{r,s^-}(t+1)=x_{r,s^+}(t)$ and $x_{r,s^+}(t+1)=x_{r,s^-}(t)$, i.e. $x_{r,4}(t+1)=x_{r,6}(t)=1$ and $x_{r,6}(t+1)=x_{r,4}(t)=2$

Step 8. $\mathrm{S}=\mathrm{S}\setminus\{s^+,s^-\}=\{4,6,\}\setminus\{4,6\}=\{\varnothing\}$, i.e. $|\mathrm{S}|=0$ and hence stop.

The stopping criterion of DE could be a given number of consecutive iterations within which no improvement on solutions, a specified CPU time limit, or maximum number of iterations (fitness calculation), $t_{\max}$, is attained. Unless otherwise specified, in this paper we use the last one as the termination criteria, i.e. the algorithm terminates when a maximum number of fitness calculation is achieved.

Extractive summarization works by choosing a subset of the sentences in the original document. This process can be viewed as identifying the most salient sentences in a cluster that give the necessary and sufficient amount of information related to main content of the cluster (topic). In a cluster of related sentences, many of the sentences are expected to be somewhat similar to each other since they are all about the same topic. The approach, proposed in papers [13,19], is to assess the centrality of each sentence in a cluster and extract the most important ones to include in the summary. In centroid-based summarization, the sentences that contain more words from the centroid of the cluster are considered as central. Centrality of a sentence is often defined in terms of the centrality of the words that it contains. In this section we use other criterion to assess sentence salience, developed in [11] which is based on technique proposed in [47].

5. Experiments and Results

For evaluation the performance of our methods we used two document datasets DUC2001 and DUC2002 and corresponding 100-word summaries generated for each of documents. The DUC2001 and DUC2002 are an open benchmark datasets which contain 309 and 567 documents-summary pairs from Document Understanding Conference (http://duc.nist.gov). The datasets DUC2001 and DUC2002 are clustered into 30 and 59 topics, respectively.

At a preprocessing step, the stopwords in each sentence were removed using the stoplist provided in ftp://ftp.cs.cornell.edu/pub/smart/english.stop and the remaining words were stemmed using the Porter's scheme [48].

We use the ROUGE (Recall Oriented Understudy for Gisting Evaluation) toolkit [49,50], which was adopted by DUC for automatically summarization evaluation. It has been shown that ROUGE is very effective for measuring document summarization. It measures summary quality by counting overlapping units such as the N-gram, word sequences and word pairs between the candidate summary and the reference summary. The ROUGE-N measure compares N-grams of two summaries, and counts the number of matches. The measure is defined by Formula (31) [49–51]:

$$\text{ROUGE-N}=\frac{\sum_{S\in Summ_{ref}}\sum_{\text{N-gram}\in S} Count_{match}(\text{N-gram})}{\sum_{S\in Summ_{ref}}\sum_{\text{N-gram}\in S} Count(\text{N-gram})} \quad (14)$$

where N stands for the length of the N-gram, $Count_{match}(\mathrm{N-gram})$ is the maximum number of N-grams co-occurring in candidate summary and a set of reference-summaries. $Count(\mathrm{N-gram})$ is the number of N-grams in the reference summaries. We show three of the ROUGE metrics in the experimental results: ROUGE-1, ROUGE-2 and ROUGE-SU4. The ROUGE-1 and ROUGE-2 scores are based on the overlap of unigrams and bigrams, respectively, between the candidate summary and the reference summary. The ROUGE-SU4 score is also based on the overlap of bigrams between summaries, but allows for gaps to occur between words (skip-bigram), with a maximum gap length of words, and includes unigram co-occurrence statistics as well.

The optimization procedure used here is stochastic in nature. Hence, for each criterion function (F_1, F_2 and F) it has been run several times for different values of parameters $CR\in[0.3;0.8]$ and $MR\in[0.1;0.5]$. At experiments the size of population and the number of iteration we kept unchanged changing only parameters *CR* and *MR* with step 0.1. For both datasets we take the same number of iterations which is 1000. The population size

is 40 % of the total number of documents in the datasets. The parameters of the DE are reported in Table 1.

The first experiment compares our methods with other methods. We compare our proposed methods with both supervised and unsupervised methods. Among the supervised methods we choose *SVM* [4] and *CRF* [21]. *SVM* is one of the state-of the-art classifiers. *CRF* combines the merits of *HMM* (Hidden Markov Model) and *LR* (Logistic Regression). *HMM* extends Naive Bayes (*NB*) by considering the sequential information, while *LR* is a discriminative version of *NB*. The unsupervised methods we compare include *QCS* [16] and graph-based algorithm *HITS* [27] (Among the several options of graph-based algorithm [27] the method based on the authority score of HITS on the directed backward graph is the best. Therefore it is taken by us for comparison). Table 2 and 3 show the results of all the methods in terms ROUGE-1, ROUGE-2, and ROUGE-SU4 metrics on DUC2001 and DUC2002 datasets, respectively. As shown in Tables 2 and 3, on DUC2001 dataset, the values of ROUGE-1, ROUGE-2 and ROUGE-SU4 metrics of all the methods are better than on DUC2002 dataset. In the Tables 2 and 3 highlighted (***bold italic***) entries represent the best performing methods.

The numerical comparison of our methods with the methods *CRF, QCS, HITS* and *SVM* is shown in Tables 4 and 5. Here we use relative improvement $\frac{(\text{our method} - \text{other methods})}{\text{other methods}} \times 100$ for comparison. In spite of the fact that among our criterion functions the worst result is obtained by criterion function F_1 but it shows better result than the other methods.

Compared with the best method *QCS* on DUC2001 (DUC2002) dataset the criterion function F_1 improves the performance by 1.05% (0.57%), 0.37% (0.43%) and 0.59% (0.31%) in terms ROUGE-1, ROUGE-2 and ROUGE-SU4 metrics, respectively.

Table 1. Parameters of the DE

Dataset	***Population size,*** N	***Number of iteration,*** $t_{\max}$	***Crossover rate,*** CR	***Mutation rate,*** MR
DUC2001	155	1000	0.6	0.2
DUC2002	285	1000	0.6	0.2

Table 2. ROUGE scores for summarization methods on DUC2001 dataset

Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
F	0.45836	0.19164	0.21574
F_1	0.45603	0.19046	0.21427
F_2	***0.45952***	***0.19338***	***0.21763***
CRF	0.44598	0.18564	0.20934
QCS	0.45129	0.18976	0.21302
HITS	0.43528	0.18317	0.20627
SVM	0.43132	0.18136	0.20372

Table 3. ROUGE scores for summarization methods on DUC2002 dataset

Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
F	0.45119	0.18847	0.21184
F_1	0.44985	0.18896	0.21234
F_2	***0.45412***	***0.18982***	***0.21268***
CRF	0.44155	0.17974	0.20129
QCS	0.44865	0.18766	0.21119
HITS	0.42806	0.16792	0.18924
SVM	0.43405	0.17084	0.19036

Table 4. Comparison our methods with other methods on DUC2001 dataset

Methods	Metrics	F_2	F_1	F
		% of improvement		
CRF	ROUGE-1	3.04	2.25	2.78
	ROUGE-2	4.17	2.60	3.23
	ROUGE-SU4	3.96	2.36	3.06
QCS	ROUGE-1	1.82	1.05	1.57
	ROUGE-2	1.91	0.37	0.99
	ROUGE-SU4	2.16	0.59	1.28
HITS	ROUGE-1	5.57	4.77	5.30
	ROUGE-2	5.57	3.98	4.62
	ROUGE-SU4	5.51	3.88	4.59
SVM	ROUGE-1	6.54	5.73	6.27
	ROUGE-2	6.63	5.02	5.67
	ROUGE-SU4	6.83	5.18	5.90

Table 5. Comparison our methods with other methods on DUC2002 dataset

Methods	Metrics	F_2	F_1	F
		% of improvement		
CRF	ROUGE-1	2.85	2.18	1.88
	ROUGE-2	5.61	4.86	5.13
	ROUGE-SU4	5.66	5.24	5.49
QCS	ROUGE-1	1.22	0.57	0.27
	ROUGE-2	1.15	0.43	0.69
	ROUGE-SU4	0.71	0.31	0.54
HITS	ROUGE-1	6.09	5.40	5.09
	ROUGE-2	13.04	12.24	12.53
	ROUGE-SU4	12.39	11.94	12.21
SVM	ROUGE-1	4.62	3.95	3.64
	ROUGE-2	11.11	10.32	10.61
	ROUGE-SU4	11.73	11.28	11.55

6. Conclusions

We have presented an unsupervised approach to automatic document summarization. Our approach consists of two steps. First sentences are clustered, and then representative sentences are defined and extracted on each cluster. In our study we developed a modified discrete differential evolution algorithm to optimize the objective functions. When comparing our methods to several existing summarization methods on an open DUC2001 and DUC2001 datasets, we found that our methods can improve the summarization results significantly. The methods were evaluated using ROUGE-1, ROUGE-2 and ROUGE-SU4 metrics.

REFERENCES

[1] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," Computer Speech and Language, Vol. 23, No. 1, pp.

126–144, 2009.

[2] U. Hahn and I. Mani, "The challenges of automatic summarization," IEEE Computer, Vol. 33, No. 11, pp. 29–36, 2000.

[3] I. Mani and M. T. Maybury, "Advances in automated text summarization," MIT Press, Cambridge, 442p, 1999.

[4] J-Y. Yeh, H-R. Ke, W-P. Yang, I-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," Information Processing and Management, Vol. 41, No. 1, pp. 75–95, 2005.

[5] S. Ye, T.-S. Chua, M.-Y. Kan, and L. Qiu, "Document concept lattice for text understanding and summarization, Information Processing and Management, 2007, Vol. 43, No. 6, pp. 1643–1662.

[6] R. M. Alguliev and R. M. Aliguliyev, "Effective summarization method of text documents," Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), France, pp. 264–271, 19–22 September 2005.

[7] R. M. Alguliev and R. M. Alyguliev, "Automatic text documents summarization through sentences clustering," Journal of Automation and Information Sciences, Vol. 40, No. 9, pp. 53–63, 2008.

[8] R. M. Alguliev, R. M. Aliguliyev, and A. M. Bagirov, "Global optimization in the summarization of text documents," Automatic Control and Computer Sciences, Vol. 39, No. 6, pp. 42–47, 2005.

[9] R. M. Aliguliyev, "A novel partitioning-based clustering method and generic document summarization," Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'06 Workshops) (WI-IATW'06), Hong Kong, China, pp. 626–629, 18–22 December 2006.

[10] R. M. Aliguliyev, "A new sentence similarity measure and sentence based extractive technique for automatic text summarization," Expert Systems with Applications, Vol. 36, No. 4, pp. 7764–7772, 2009.

[11] R. M. Aliguliyev, "Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization, Computational Intelligence (accepted).

[12] Y. Gong and 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, New Orleans, USA, pp. 19–25, 9–12 September, 2001.

[13] D. R. Radev, H. Jing, M. Stys, and D. Tam, "Centroid-based summarization of multiple documents," Information Processing and Management, Vol. 40, No. 6, pp. 919–938, 2004.

[14] G. Salton, A. Singhal, M. Mitra, and C. Buckley, "Automatic text structuring and summarization," Information Processing and Management, Vol. 33, No. 2, pp. 193–207, 1997.

[15] K. M. Svore, L. Vanderwende, and C. J. C. Burges, "Enhancing single-document summarization by combining RankNet and third-party sources," in: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07), Prague, Czech Republic, pp. 448–457, 28–30 June 2007.

[16] D. M. Dunlavy, D. P. O'Leary, J. M. Conroy, and J. D. Schlesinger, "QCS: A system for querying, clustering and summarizing documents," Information Processing and Management, Vol. 43, No. 6, pp. 1588–1605, 2007.

[17] K. S. Jones, "Automatic summarizing: the state of the art," Information Processing and Management, Vol. 43, No. 6, pp. 1449–1481, 2007.

[18] D. Zajic, B. J. Dorr, J. Lin, and R. Schwartz, "Multi-candidate reduction: sentence compression as a tool for document summarization tasks," Information Processing and Management, Vol. 43, No. 6, pp. 1549–1570, 2007.

[19] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," Journal of Artificial Intelligence Research, Vol. 22, pp. 457–479, 2004.

[20] D. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization," Computational Linguistics, Vol. 28, No. 4, pp. 399–408, 2002.

[21] D. Shen, J.-T.Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, pp. 2862–2867, January 6–12, 2007.

[22] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," Proceedings of the 18th International Conference on Machine Learning, pp. 282–289, 28 June–01 July 2001.

[23] X. Wan, "A novel document similarity measure based on earth mover's distance," Information Sciences, Vol. 177, No. 18, pp. 3718–3730, 2007.

[24] S. Fisher and B. Roark, "Query-focused summarization by supervised sentence ranking and skewed word distributions," Proceedings of the Document Understanding Workshop (DUC'06), New York, USA, 8p, 8–9 June 2006.

[25] P. Fung and G. Ngai, "One story, one flow: Hidden Markov story models for multilingual multidocument summarization," ACM Transaction on Speech and Language Processing, Vol. 3, No. 2, pp. 1–16, 2006.

[26] D. M. McDonald and H. Chen, "Summary in context: Searching versus browsing," ACM Transactions on Information Systems, Vol. 24, No. 1, pp. 111–141, 2006.

[27] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarizations," Proceedings of the Second International Joint Conference Natural Language Processing (IJCNLP'05), Korea, pp. 602–607, 11–13 October 2005.

[28] J. Li, L. Sun, C. Kit, and J. Webster, "A query-focused multi-document summarizer based on lexical chains,"

Proceedings of the Document Understanding Conference (DUC'07), New York, USA, 4p, 26–27 April 2007.

[29] X. Wan, "Using only cross-document relationships for both generic and topic-focused multi-document summarizations, Information Retrieval, Vol. 11, No. 1, pp. 25–49, 2008.

[30] R. Mihalcea and H. Ceylan, "Explorations in automatic book summarization, Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07), Prague, Czech Republic, pp. 380–389, 28–30 June 2007.

[31] Y. Guo and G. Stylios, "An intelligent summarization system based on cognitive psychology," Information Sciences, Vol. 174, No.1–2, pp. 1–36, 2005.

[32] J. Grabmeier and A. Rudolph, "Techniques of cluster algorithms in data mining," Data Mining and Knowledge Discovery, Vol. 6, No. 4, pp. 303–360, 2002.

[33] J. Han and M. Kamber, "Data mining: concepts and technique (2nd Edition), Morgan Kaufman, San Francisco, 800p, 2006.

[34] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," ACM Computing Surveys, Vol. 31, No. 3, pp. 264–323, 1999.

[35] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "An overview of clustering methods," Intelligent Data Analysis, Vol. 11, No. 6, pp. 583–605, 2007.

[36] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, pp. 1279–1296, 2004.

[37] T. Li, "A unified view on clustering binary data," Machine Learning, Vol. 62, No. 3, pp. 199–215, 2006.

[38] Y. Li, C. Luo and S. M. Chung, "Text clustering with feature selection by using statistical data," IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 20, pp. 641–652, 2008.

[39] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 8, pp. 1138–1150, 2006.

[40] X. Liu, Y. Zhou, and R. Zheng, "Sentence similarity based on dynamic time warping," Proceedings of the First International Conference on Semantic Computing (ICSC' 07), Irvine, USA, pp. 250–256, 17–19 September 2007.

[41] X. Wan, J. Yang, and J. Xiao, "Manifold-ranking based topic-focused multi-document summarization, Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, pp. 2903–2908, January 6–12, 2007.

[42] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," Proceedings of 16th World Wide Web Conference (WWW16), Alberta, Canada, pp. 757–766, May 8–12, 2007.

[43] R. L. Cilibrasi and P. M. B. Vitanyi, "The google similarity distance," IEEE Transaction on Knowledge and Data Engineering, Vol. 19, No. 3, pp. 370–383, 2007.

[44] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," Machine Learning, Vol. 55, No. 3, pp. 311–331, 2004.

[45] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," IEEE Transaction on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 38, No. 1, pp. 218–237, 2008.

[46] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, Vol. 11, No. 4, pp. 341–359, 1997.

[47] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 1, pp. 167–172, 2007.

[48] M. Porter, "An algorithm for suffix stripping," Program, Vol. 14, No. 3, pp. 130–137, 1980.

[49] C.-Y. Lin, "ROUGE: A package for automatic evaluation summaries," Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, pp. 74–81, 25–26 July 2004.

[50] C.-Y. Lin and E. H. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL'03), Edmonton, Canada, Vol. 1, pp. 71–78, 27 May–1 June 2003.

[51] H. Nanba and M. Okumura, "An automatic method for summary evaluation using multiple evaluation results by a manual method," Proceedings of the COLING/ACL on Main Conference Poster Sessions, Sydney, Australia, pp. 603–610, 17–18 July 2006.

