International Journal of Intelligent Information Technologies
Volume 13 • Issue 1 • January-March 2017

A Model for Text Summarization

Rasim M. Alguliyev, Azerbaijan National Academy of Sciences, Institute of Information Technology, Baku, Azerbaijan

Ramiz M. Aliguliyev, Azerbaijan National Academy of Sciences, Institute of Information Technology, Baku, Azerbaijan

Nijat R. Isazade, Azerbaijan National Academy of Sciences, Institute of Information Technology, Baku, Azerbaijan

Asad Abdi, University of Malaya, Department of Artificial Intelligence, Kuala Lumpur, Malaysia

Norisma Idris, University of Malaya, Department of Artificial Intelligence, Kuala Lumpur, Malaysia

ABSTRACT

Text summarization is a process for creating a concise version of document(s) preserving its main content. In this paper, to cover all topics and reduce redundancy in summaries, a two-stage sentences selection method for text summarization is proposed. At the first stage, to discover all topics the sentences set is clustered by using k-means method. At the second stage, optimum selection of sentences is proposed. From each cluster the salient sentences are selected according to their contribution to the topic (cluster) and their proximity to other sentences in cluster to avoid redundancy in summaries until the appointed summary length is reached. Sentence selection is modeled as an optimization problem. In this study, to solve the optimization problem an adaptive differential evolution with novel mutation strategy is employed. With a test on benchmark DUC2001 and DUC2002 data sets, the ROUGE value of summaries got by the proposed approach demonstrated its validity, compared to the traditional methods of sentence selection and the top three performing systems for DUC2001 and DUC2002.

KEYWORDS

Adaptive Differential Evolution Algorithm, Information Diversity, k-Means, Optimization Model, Sentence Clustering, Text Summarization

INTRODUCTION

Interest in text summarization has gained increasing attention in recent years because of the large amounts of text data, which are created in a variety of social networks, web, and other information-centric applications, such as e-library and e-government. The explosion of electronic documents has made it difficult for users to extract useful information from them. The user due to the large amount of information does not read many relevant and interesting documents. Therefore, the continuing growth of available online text documents makes research and application of text summarization very important and consequently attracts many researchers. The reason for this is twofold: first, text summarization can help cope with the information overload, and second, small form-factor devices are becoming increasingly popular.

Text summarization is a process of automatically creating a shorter version of a document or a set of documents by reducing the document(s) in length. It is an important way of finding relevant information in large text libraries or in the Internet (Canhasi & Kononeko, 2014; Ferreira et al., 2014).

DOI: 10.4018/IJIIT.2017010104

Text summarization can help users to access the information more easily, on the one hand, reducing the time they have to spend dealing with the information, and on the other, selecting the information most useful for them (Yang & Wang, 2008; Lloret & Palomar, 2013).

According to different criteria, text summarization techniques can be categorized into abstract-based and extract-based (reproducing sentence or not), multi-document and single-document (more than one document or not), query-focused and generic (given query or not), supervised and unsupervised (with training set or not) methods. Abstraction can be described as reading and understanding the text to recognize its content which is then compiled in a concise text. In general, an abstract can be described as summary comprising concepts/ideas taken from the source that are then reinterpreted and presented in a different form. An extract is a summary consisting of units of text taken from the source and presented verbatim. Single-document summarization can only distill one document into a shorter version, while on the contrary; multi-document summarization can compress a set of documents. Multi-document summarization can be seen as an enhancement of single-document summarization and can be used for outlining the information contained in a cluster of documents (Canhasi & Kononeko, 2014; Luo, Zhuang, He, & Shi, 2013). Generic summarization tries to extract the most general idea from the original document without any specified preference in terms of content. Query-focused document summarization is a special case of document summarization. Given a query, the task is to produce a summary which can respond to the information required by the query (Canhasi & Kononeko, 2014). In supervised methods for summarization, the task of selecting important sentences is represented as a binary classification problem, partitioning all sentences in the input into summary and non-summary sentences. Unsupervised learning methods do not require any training data, thus can be applied to any text data without requiring any manual effort. The two main unsupervised learning methods commonly used in the context of text data are clustering and topic modeling (Aliguliyev, 2010; Cai, Li, & Zhang, 2013; Cai, Li, & Zhang, 2014; Cai, Li, Zhang, & Shi, 2014; Mei & Chen, 2012).

In this paper, we focus on unsupervised, i.e., on clustering and optimization based extractive document summarization. For detecting topics in a document collection this approach, firstly, utilizes clustering approach to segment the sentences into topical groups. Secondly, to form an optimal summary this approach presents an optimization model to select the representative sentences from each group. Later, to solve the optimization problem a modified differential evolution algorithm is developed. Notice that this approach allows avoiding redundancy in a creating summary and covering all topics in the document collection. The proposed method is called Generic Document Summarization based on Clustering and Optimization (GDSCO).

The rest of this paper is organized as follows. Section 2 introduces the overview of related work. In Section 3 mathematical formulation of sentence selection problem for text summarization is introduced. It firstly segregates the sentences into clusters by topics and then the sentence selection problem from each cluster is formulated as an optimization problem. Section 4 describes a modified DE algorithm for solving the optimization problem. Finally, we conclude our paper in Section 5.

RELATED WORK

There are many methods to summarize documents by finding topics of the document first and scoring the individual sentences with respect to the topics. Sentence clustering has been successfully applied in document summarization to discover the topics conveyed in a document collection. However, existing clustering-based summarization approaches are seldom targeted for both diversity and coverage of summaries, which are believed to be the two key issues to determine the quality of summaries (Cai

& Li, 2011). The focus of the work (Cai, Li, & Zhang, 2014) is to explore a systematic approach that allows diversity and coverage to be tackled within an integrated clustering-based summarization framework. Cai et al. (2013) developed two co-clustering frameworks, namely integrated clustering and interactive clustering, to cluster sentences and words simultaneously. Co-clustering frameworks are proposed to allow words to play an explicit role in sentence clustering as an independent text object and to allow simultaneous sentence and word clustering. A fuzzy medoid-based clustering approach for query-oriented multi-document summarization, presented in (Mei & Chen, 2012), is successfully employed to generate subsets of sentences where each of them corresponds to a subtopic of the related topic. For detecting relevant information and avoiding redundant information in the summaries Lloret and Palomar (2013) presented a text summarization tool, called compendium. It combines the statistical and cognitive-based techniques for detecting relevant information and for avoiding redundant information it uses textual entailment. Luo et al. (2013) proposed a probabilistic-modeling relevance, coverage, and novelty framework to model topic relevance and coverage, where a reference topic model incorporating query is utilized for dependent sentence relevance measurement.]

In (Ferreira et al., 2014), to avoid information redundancy and to provide diversity in a summary, a new sentence clustering algorithm based on a graph model that makes use of statistic similarities and linguistic treatment is proposed. Multi-document summarization is also carried out using graph-based approaches, such as (Erkan & Radev, 2004), (Wan, Yang, & Xiao, 2007) and (Balaji, Geetha, & Parthasarathi, 2014). LexRank (Erkan & Radev, 2004) builds a graphical representation of a document. Then the LexRank algorithm ranks the graph nodes in terms of their centrality, i.e. the most connected nodes are ranked highest. CollabSum (Wan, Yang, & Xiao, 2007) first employs the clustering algorithm to obtain appropriate document clusters and then utilizes the graph-based ranking algorithm for collaborative document summarization within each cluster. Both the cross-document and the within-document relationships between sentences are incorporated in the algorithm. The within-document relationships reflect the local information existing in the document and the cross-document relationships reflect the global information in the cluster context.

The use of optimization models for summarization purposes has also been investigated by many researchers. For example, in (Alguliev, Aliguliyev, & Hajirahimova, 2012; Alguliev, Aliguliyev, & Isazade, 2013a; Alguliev, Aliguliyev, & Isazade, 2013b) the authors formalized the sentence selection task as an optimization problem and solved the problem by using evolutionary and swarm optimization algorithms. A method, called MCLR (Maximum Coverage and Less Redundancy) (Alguliev, Aliguliyev, & Hajirahimova, 2012), document summarization models as a quadratic Boolean programming problem where objective function is a weighted combination of the content coverage and redundancy objectives. Another successful constraint-driven document summarization model is presented by Alguliev, Aliguliyev, and Isazade (2013a) where the document summarization is modeled as a quadratic integer programming problem and solved with discrete binary particle swarm optimization algorithm. In (Takamura & Okumura, 2009), text summarization modeled as a maximum coverage problem that aims at covering as many conceptual units as possible by selecting some sentences. Nishino et al. (2013) formalized the extractive text summarization task as a combinatorial optimization problem of maximizing an objective function that measures summary quality. The objective function combines the three objectives of relevance, redundancy, and coverage. Authors introduced Lagrangian relaxation based heuristics for obtaining a good approximation solution in much shorter time than is possible with integer linear programming.

MATHEMATICAL MODEL

Problem Statement

Given a document collection $D = \{D_1, ..., D_N\}$, where N is the number of documents. For simplicity, we represent the document collection as the set of all sentences from all the documents in the collection, i.e. $S = \{S_1, \ldots, S_n\}$, where S_i denotes ith sentence in D, n is the number of sentences in the document collection. We attempt to find a subset of the sentences $S = \{S_1, \ldots, S_n\}$ that covers the different topics of the document collection while reducing the redundancy in the summary.

Generally, a document contains a variety of information centered on a main theme, and covering different aspects of the main topic. Coverage means that the generated summary should cover all subtopics as much as possible. Poor subtopics coverage is usually manifested by absence of some summary sentences. Therefore, when doing summarization, if only focusing on the sentences with higher relevance scores to the whole document, the summary sentences extracted are inclined to sentences in the subtopics whose sentences distribute widely. Moreover, the subtopics whose sentences do not distribute widely will be ignored. For this reason, when extracting summary sentences, we not only focus on the relevance scores of sentences to the whole sentence collection, but also the topic representative of sentences. The summary sentences should include most of all the subtopics.

In our study, we segment a sentence collection according to its topics. To segment the sentence collection into subtopics we use the *k*-means algorithm. When generating a summary, we also need to deal with the problem of repetition of information. This problem is especially important for multi-document summarization, where multiple documents will discuss the same topic. It is known that each of the selected sentences included in the summary should be individually important. However, this does not guarantee they collectively produce the best summary. For example, if the selected sentences overlap a lot with each other, such a summary is definitely not desired. When many of the competing sentences are available, given summary length limit, the strategy of selecting best summary rather than selecting best sentences becomes evidently important. Therefore, selecting the best summary is a global optimization problem in comparison with the procedure of selecting the best sentences.

The Similarity Measure

Let $T = \{t_1, t_2, \ldots, t_m\}$ represents all the distinct terms occurred in the document collection D, where m is the number of terms. According to the vector space model each sentence s_i is represented using these terms as a vector in m-dimensional space, $S_i = [w_{i1}, \ldots, w_{im}]$, $i = 1, \ldots, n$, where each component reflects weight of a corresponding term. Different weighting schemes are available. The common and popular one is the Term Frequency–Inverse Document Frequency (TF-IDF) weighting scheme. In this study, instead of using simple *tf-isf* (term frequency–inverse sentence frequency) scheme, symmetric Okapi BM25 (Song, Liang, & Park, 2014) framework is utilized for indexing term weights:

$$w_{ij} = \frac{tf_{ij}}{tf_{ij} + 0.5 + 1.5 \times \frac{l_i}{avgl}} \times isf_j \tag{1}$$

where inverse sentence frequency *isf* is obtained by dividing the total number of sentences by the number of sentences containing the term, and then taking the logarithm of that quotient:

$$isf_j = \log\left(\frac{n}{n_j}\right) \tag{2}$$

Here n is the total number of sentences in the document collection D; n_j is the number of sentences in which the term t_j occurred; tf_{ij} is the number of occurrences of term t_j in sentence

S_i, l_i is the length of sentence S_i and $avgl$ is the average sentence length. This formula normalizes the length of sentences rather than the simple *tf-isf* method.

The mining similarity measure plays an important role. Intuitively, if there are many common words between two sentences, they are very similar. Given two sentences $S_i = \left[w_{i1}, \ldots, w_{im}\right]$ and $S_j = \left[w_{j1}, \ldots, w_{jm}\right]$. To measure similarity between two sentences we use the following measure (Alguliyev, Aliguliyev, & Isazade, 2015):

$$sim_{RRN}\left(S_i, S_j\right) = 1 - \frac{2 \cdot \sum_{k=1}^{m}\left(w_{ik} - w_{ik}w_{jk}\right) \cdot \sum_{k=1}^{m}\left(w_{jk} - w_{ik}w_{jk}\right)}{\sum_{k=1}^{m} w_{jk} \cdot \sum_{k=1}^{m}\left(w_{ik} - w_{ik}w_{jk}\right) + \sum_{k=1}^{m} w_{ik} \cdot \sum_{k=1}^{m}\left(w_{jk} - w_{ik}w_{jk}\right)} \tag{3}$$

Clustering Stage

In this subsection, the sentences are clustered into different groups to discover latent subtopic information in the document collection. Generally, automatic clustering is a process of dividing a set of objects into unknown groups, where the clustering algorithm determines the best number k of groups (or clusters). 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.

Clustering is a popular exploratory pattern classification technique which partitions the input data into k groups based on some similarity/dissimilarity metric, where the value of k may or may not be known a priori. The main objective of any clustering technique is to produce a $k \times n$ partition matrix $U(X)$ of the given data set X, consisting of n patterns, $X = \left\{x_1, x_2, \ldots, x_n\right\}$. The partition matrix may be represented as $U = \left[u_{iq}\right]$ ($i = 1, 2, \ldots, n$ and $q = 1, 2, \ldots, k$) where u_{iq} is the membership of pattern x_i to the qth cluster. For fuzzy clustering of the data, $0 < u_{iq} < 1$, i.e., u_{iq} denotes the degree of belongingness of pattern x_i to the qth cluster. For hard clustering of the data $u_{iq} \in \{0, 1\}$.

We consider the hard unconstrained partition clustering problem, that is the distribution of the sentences of the set $S = \left\{s_1, \ldots, s_n\right\}$ into a given number k of disjoint subsets C_q, $q = 1, 2, \ldots, k$, with respect to predefined criteria such that:

For any $q = 1, 2, \ldots, k$ $C_q \neq \varnothing$, i.e. each cluster should have at least one sentence assigned.

For any $q1 \neq q2$ $C_{q1} \cap C_{q2} = \varnothing$, $q1, q2 = 1, 2, \ldots, k$, i.e. two different clusters should have no sentences in common:

$$\bigcup_{q=1}^{k} C_q = S$$

i.e. each sentence should definitely be attached to a cluster.

No constraints are imposed on the clusters C_q, $q = 1, 2, \ldots, k$.

The sets C_q, $q = 1, 2, \ldots, k$ are called clusters. We assume that each cluster C_q can be identified by its center $O_q \in \mathrm{R}^m$, $q = 1, 2, \ldots, k$.

The k-means algorithm is formally defined as follows:

Step 1: Let k be the number of clusters. In this study, it is defined by Equation 8.

Step 2: Initialize the centers to k random locations in the collection $S = \{S_1, \ldots, S_n\}$ and calculate the mean center of each cluster, O_q, where O_q is the center of cluster C_q.

Step 3: Calculate the similarity from the center of each cluster to each input sentence vector, assign each input sentence vector to the cluster where the similarity between itself and O_q is maximal. Recompute O_q for all clusters that have inherited a new input sentence vector, and update each cluster center (if there are no changes within the cluster centers, discontinue recomputation).

Step 4: Repeat Step 3 until all the sentences are assigned to their optimal cluster centers. This ends the cluster updating procedure with k disjoint subsets.

There are different reformulations of the clustering problem as an optimization problem. The k-means algorithm is based on a within-class compactness, which measures the similarity between input vectors S, and cluster representatives O_q using the objective function (Bagirov, Ugon, & Webb, 2011):

Maximize:

$$\sum_{q=1}^{k} \sum_{i=1}^{n} sim_{RRN}(S_i, O_q) u_{iq} \tag{4}$$

subject to:

$$\sum_{q=1}^{k} u_{iq} = 1, \ \forall i \tag{5}$$

$$1 < \sum_{i=1}^{n} u_{iq} < k, \ \forall q \tag{6}$$

$$u_{iq} \in \{0,1\}, \ \forall i, q \tag{7}$$

where:

$$u_{iq} = \begin{cases} 1, & if \ S_i \in C_q \\ 0, & otherwise \end{cases}$$

$O_q = \left[w_1^q, \ldots, w_m^q\right]$ is the center of cluster C_q, l th coordinate w_l^q of which is calculated as:

$$w_l^q = \frac{1}{\left|C_q\right|} \sum\nolimits_{i=1}^{n} w_{il} u_{iq}$$

$|C_q|$ is the number of sentences assigned to cluster C_q; $sim_{RRN}\left(S_i, O_q\right)$ is the similarity measure between $S_i = \left[w_{i1}, \ldots, w_{im}\right]$ and $O_q = \left[w_1^q, \ldots, w_m^q\right]$.

In text clustering the latent topic number in the document collection cannot be predicted, so it is impossible to offer k effectively. The strategy that we used to determine the optimal number of clusters (the number of topics in a document) is based on the distribution of words in the sentences (Alguliyev, Aliguliyev, & Isazade, 2015):

$$k = n\frac{\left|\bigcup_{i=1}^{n} S_i\right|}{\sum_{i=1}^{n}\left|S_i\right|} \tag{8}$$

where $|S_i|$ is the number of terms in the sentence S_i.

In other words, the number of clusters (i.e. the number of topics in a document collection) is defined as n times the ratio of the total number of terms in the document collection to the cumulative number of terms in the sentences considered separately.

Optimization Stage

We formalize the text summarization problem as the optimization problem of maximizing an objective function that measures summary quality. Our objective function combines the two objectives of relevance (relevance of a summary is the amount of relevant information the summary contains) and redundancy (summary should not contain multiple sentences that convey the same information):

Maximize:

$$f\left(X\right) = \sum_{q=1}^{k}\sum_{i=1}^{n} sim_{RRN}\left(S_i, O_q\right)x_{iq} + \sum_{q=1}^{k}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(1 - sim_{RRN}\left(S_i, S_j\right)\right)x_{iq}x_{jq} \tag{9}$$

subject to:

$$\sum_{q=1}^{k}\sum_{i=1}^{n} l_i x_{iq} \le L_{\max} \tag{10}$$

$$x_{iq} \in \{0,1\},\ \forall i \tag{11}$$

Here x_{iq} denotes a variable which is 1 if sentence S_i from cluster C_q is selected to be included to the summary, otherwise 0. $L_{\max}$ is the limit length of the summary, l_i denotes the length of sentence S_i.

Equation 10 is the cardinality constraint, which guarantees that the summary is bounded in length. The integrality constraint on x_{iq} (Equation 11) is automatically satisfied in the problem above. Now our objective is to find the binary assignment $X = \{x_{iq}\}$ (Equation 11) with the best content coverage and less redundancy (Equation 9) such that the summary length is at most $L_{\max}$ (Equation 10).

The objective Equation 9 balances the content coverage and diversity in the summary. The first term aims to evaluate the wide content coverage of the summary. The high value of the term provides that sentences be well grouped in groups according to topics. As said above the summary should not contain multiple sentences that convey the same information. Therefore, at choosing of sentences as a candidate sentence of summary, it is necessary to meet a condition that similarity between selected sentences is minimized. This requirement provides the second term. The second term minimizes the sum of inter-sentence similarities among sentences chosen from S. A higher value of this term corresponds to higher diversity in the summary.

AN ADAPTIVE DIFFERENTIAL EVOLUTION WITH A NOVEL MUTATION STRATEGY

Many techniques can be used to solve the optimization problems (4)-(7) and (9)-(11). In recent years, a new optimization method known as differential evolution (DE) has gradually become more popular and has been successfully applied to solve many optimization problems (Storn & Price, 1997; Das & Suganthan, 2011; Cheng, Zhang, & Neri, 2013; Li, Kwong, & Deb, 2015; Rakshit & Konar, 2015). In our study, the optimization problems (9)-(11) was solved using a DE algorithm.

The DE algorithm is a population-based algorithm like genetic algorithms using the three operators: crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operation. This main operation is based on the differences of randomly sampled pairs of solutions in the population. The algorithm uses mutation operation as a search mechanism and selection operation to direct the search toward the prospective regions in the search space.

The basic idea which DE scheme is based on is to generate new trial vector. When mutation is implemented, several differential vectors obtained from the difference of several randomly chosen parameter vectors are added to the target vector to generate a mutant vector. Then, a trial vector is produced by crossover recombining the obtained mutant vector with the target vector. Finally, if the trial vector yields better fitness value than the target vector, replace the target vector with the trial vector. The main steps of the basic DE algorithm are described below.

Encoding of the Chromosomes and Population Initialization

The basic DE (Storn & Price, 1997; Das & Suganthan, 2011) is a population-based global optimization method that uses a real-coded representation. Like the other evolutionary algorithms, DE also starts with a population of P n-dimensional search variable vectors. The pth individual vector of the population at generation t has n components, $U_p(t) = \left[u_{p,1}(t), \ldots, u_{p,n}(t)\right]$, where $u_{p,s}(t)$ is the sth decision variable of the pth chromosome in the population, $s = 1, 2, \ldots, n$; $p = 1, 2, \ldots, P$.

These vectors are referred in literature as "genomes" or "chromosomes". In the initialization procedure, P solutions will be created at random to initialize the population. At the very beginning of a DE run, problem independent variables are initialized in their feasible numerical range. Therefore, if the sth variable of the given problem has its lower and upper bound as $u_s^{\min}$ and $u_s^{\max}$, respectively, then the sth component of the pth population member $U_p(t)$ may be initialized as:

$$u_{p,s}\left(0\right) = u_s^{\min} + \left(u_s^{\max} - u_s^{\min}\right) \cdot rand_{p,s} \tag{12}$$

where $rand_{p,s}$ is a random number between 0 and 1, chosen once for each $s \in \{1,2,\ldots,n\}$.

Modified Mutation Operator

DE is based on a mutation operator, which adds an amount obtained by the difference of two randomly chosen individuals of the current population, in contrast to most of the evolutionary algorithms, in which the mutation operator is defined by a probability function. Mutation expands the search space. In each generation to change each population member, a mutant vector is created.

For each target vector $U_p(t)$ it calculates the weighting combination of the $U_p^{lbest}(t)$, the differences $(U_p^{lbest}(t) - U_p(t))$ and $(U^{gbest}(t) - U_p(t))$, and creates a trial offspring:

$$V_p(t) = \omega U_p^{lbest}(t) + F(t) \cdot \left(U_p^{lbest}(t) - U_p(t)\right) + (1 - F(t)) \cdot \left(U^{gbest}(t) - U_p(t)\right) \quad (13)$$

where ω is the inertia weight, $U^{gbest}(t)$ is the global best solution of population and $U_p^{lbest}(t)$ is the local best solution of the pth individual during t generation, respectively, and $F(t)$ is the scaling factor:

$$F(t) = \frac{1}{1 + \exp\left(-t / t_{\max}\right)} \quad (14)$$

where t is the current generation and $t_{\max}$ is the maximum number of generations.

The inertia weight ω is linearly decreased from 0.9 to 0.4:

$$\omega(t) = 0.9 - 0.5 \frac{t}{t_{\max}} \quad (15)$$

Crossover

In order to increase the diversity of the perturbed parameter vectors, a crossover operator is introduced. The parent vector $U_p(t)$ is mixed with the mutated vector $V_p(t)$ to produce a trial vector $Z_p(t) = [z_{p,1}(t), \ldots, z_{p,n}(t)]$. It is developed from the elements of the target vector, $U_p(t)$, and the elements of the mutant vector, $Y_p(t)$:

$$z_{p,s}(t) = \begin{cases} v_{p,s}(t) & if\ rand_{p,s} \le CR\ or\ s = s^* \\ u_{p,s}(t) & otherwise \end{cases} \quad (16)$$

$CR \in [0,1]$ is the crossover constant which controls the recombination of target vector and mutant vector to generate trial vector and $s^* \in \{1,2,\ldots,n\}$ is the randomly chosen index which ensures at least one element from mutant vector is obtained by the trial vector, otherwise, there is no new vector would be produced and the population would not evolve.

Function Evaluation

The evaluation function is an operation to evaluate how good the solution (sentence selection, i.e. summary) of each individual is, making comparison between different solutions possible. The evaluation function consists of calculating the value of the objective function (9) of the summary represented by each individual.

Selection

Selection compares the quality of the trial vector $Z_p(t)$ and the target vector $U_p(t)$ and decides which one is able to survive to the next generation. To keep the population size constant over subsequent generations, the selection process is carried out to determine which one of the child and the parent will survive in the next generation, i.e., at time $t+1$. All solutions in the population have the same chance of being selected as parents without dependence of their fitness value. The child produced after the mutation and crossover operations is evaluated. Then, the performance of the child vector and of its parent is compared and the better one is selected. The target vector $\left(U_p(t)\right)$ or trial vector $\left(Z_p(t)\right)$ that generates a better solution will be selected as the target vector of the next generation $\left(U_p(t+1)\right)$. The selection formula is shown in Equation 17:

$$U_p(t+1) = \begin{cases} Z_p(t), & if\ fit\left(Z_p(t)\right) \geq fit\left(U_p(t)\right) \\ U_p(t), & otherwise \end{cases} \tag{17}$$

$fit(U)$ denotes the fitness value of individual U. Therefore, if the child yields an equal and higher value of the fitness function, it replaces its parent in the next generation; otherwise the parent is retained in the population. Hence, the population either gets better in terms of the fitness function or remains constant but never deteriorates.

Stopping Criterion

Mutation, crossover and selection continue until some stopping criterion is reached. If the predefined maximum iteration number is reached, then the DE algorithm is terminated and output the best solution obtained by DE as the result. Otherwise, it is continued to carry out individual's position updates process (mutation, crossover and selection process).

Binarization

Binary DE is the modified version of DE which operates in binary search spaces. In the binary DE, the real value of genes is converted to the binary space by the rule (Storn & Price, 1997; Das & Suganthan, 2011):

$$u_{p,s}(t+1) = \begin{cases} 1, & if\ rand_{p,s} < sigm\left(u_{p,s}(t+1)\right) \\ 0, & otherwise \end{cases} \tag{18}$$

where, as before, $rand_{p,s}$ is a uniformly distributed random number lying between 0 and 1 which is called anew for each s th component of the p th parameter vector and:

$$sigm(z) = \frac{1}{1+\exp(-z)} \tag{19}$$

is the sigmoid function.

The motivation to use the sigmoid function is to map interval $\left[u_s^{\min}, u_s^{\max}\right]$ for each $s \in \{1,2,\ldots,n\}$ into the interval $(0,1)$, which is equivalent to the interval of a probability function. After such transformation from the real-coded representation we obtain the binary-coded representation, $u_{p,s}(t) \in \{0,1\}$. Where the $u_{p,s}(t) = 1$ indicates that the s th sentence is selected to be included to the summary, otherwise, the s th sentence is not be selected. For example, the individual $U_p(t) = [1,0,0,1,1]$ represents a candidate solution that first, fourth and fifth sentences are selected to be included to the summary.

After binarization stage, we can transform the representation $U_p\left(t+1\right) = \left[u_{p,1}\left(t+1\right),\ldots,u_{p,n}\left(t+1\right)\right]$ to variables $X = \{x_{iq}\}$ used for objective function calculation (9). This transformation can be written as follow:

$$x_{ip} = \begin{cases} 1, & if\ u_{p,i} = 1 \\ 0, & otherwise \end{cases} \tag{20}$$

Constraint Handling

When population initialization, mutation, crossover and binarization have been implemented, the new generated solution may not satisfy the constraint (10). The most popular constraint handling strategy at present is penalty method, which often uses function to convert a constrained problem into an unconstraint one. Therefore, this strategy is very convenient to handle the constraints for evolutionary algorithm by punishing the infeasible solution during the selection procedure to ensure the feasible ones are favored.

To evaluate the quality of a solution provided by a chromosome, it is necessary to have a fitness function. The fitness value is an indicator of the quality of a chromosome as a solution candidate to the optimization problem under study. Therefore, in computing the value of fitness function, a penalty term is added to the fitness function in order to convert the constrained problem into an unconstrained one. An additional term is determined by penalizing the infeasible solutions with β ($\beta > 0$). Fitness function is formally defined as follows:

$$fit(X) = f(X) \cdot \exp\left(-\beta \cdot \max\left(0, \sum_{q=1}^{k}\sum_{i=1}^{n} l_i x_{iq} - L_{\max}\right)\right) \tag{21}$$

where problem variables x_{iq} are defined by the decoding rule (12).

The first multiplier $f(X)$ in Equation 21 is the objective function (9). The second multiplier is defined as an additional penalty function for maximization. β represents the cost of overloaded summary. Initial value of β is set by the user. If a solution is not feasible, the second term will be less than 1 and therefore the search will be directed to a feasible solution. If the summary length is not exceeded, this term will equal 1 to ensure the solution not to be penalized. The parameter β can be increased during the run to penalize infeasible solutions and drive the search to feasible ones that means the adaptive control of the penalty costs:

Table 1. Description of the datasets

	DUC 2001	DUC 2002
Number of clusters	30	59
Number of documents	309	567
Average number of sentences per cluster	1003	816
Data source	TREC-9	TREC-9
Task	Task 1	Task 1
Summary length	100 words	100 words

$$\beta = \beta^- + \left(\beta^+ - \beta^-\right)\frac{t}{t_{\max}} \tag{22}$$

where $t_{\max}$ is the maximum number of generations, β^- and β^+ are the start and the end values of the parameter β which we set as: $\beta^- = 0.1$ and $\beta^+ = 0.5$.

EXPERIMENTS

In this section, we conducted the experiments to assess the performance of the proposed method. We used the datasets provided in Document Understanding Conference (DUC). In the recent years, DUC http://duc.nist.gov/ has been established as a system evaluation competition for researchers to compare the performance of different summarization approaches on common datasets.

Datasets

We describe the data used throughout our experiments. We conduct experiments on the DUC2001 (http://www-nlpir.nist.gov/projects/duc/guidelines/2001.html), DUC2002 (http://www-nlpir.nist.-gov/projects/duc/guidelines/2002.html) and corresponding summaries generated for each of documents. The DUC2001 data collection contains 30 sets of approximately 10 documents from news reports in English, consisting of 309 articles that cover various topics. Each set is accompanied by reference summaries for single and multiple documents. The DUC2002 collection, meanwhile, consists of 567 documents in 59 sets. As with DUC2001, DUC2002 contains various English news articles collected from TREC-9 for the document summarization task. Table 1 gives a short summary of the datasets.

Pre-Processing

In our experiment, a pre-processing of the document is performed that contains linguistic techniques such as segmenting sentences, stop words removal, removal of upper case and stemming. The segmentation process consists of dividing the texts into meaningful units, in our experiment text are split into sentences. Using stop words removal, the words that are very common within a text and are also considered as noisy terms are removed. Obviously, their removal can be effective before the accomplishment of a natural language processing task. Such removal is usually performed by word filtering with the aid of a list of stop words. In our work, the stop words extracted from the English stop words list (http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop). Stemming is a computational procedure that reduces the words with the same root. All words are stemmed using Porter Stemmer (http://www.tartarus.org/martin/PorterStemmer/).

Evaluation Metrics

In order to evaluate and compare the performance of our proposed method, we used the standard ROUGE metric (Lin, 2004), which is adopted by DUC for automatic summarization evaluation. ROUGE measures summary quality by counting the overlapping units such as n-gram, word sequences, and word pairs between the candidate summary and a reference summary. The ROUGE system includes various automatic assessment approaches, such as ROUGE-N, ROUGE-L, and ROUGE-S. ROUGE-L calculates the similarity between a reference summary and a candidate summary based on the Longest Common Subsequence (LCS). ROUGE-S is a measure of the overlap of skip-bigrams between a candidate and a reference summary. ROUGE-SU4 (skip-bigram based with maximum skip distance of 4, plus unigram). ROUGE-N compares two summaries, the system summary and the human summary, based on total number of matches. It is calculated as follows:

$$ROUGE - N = \frac{\sum_{S \int Reference\ summaries} \sum_{N-gram\ \varepsilon\ S} Count_{match}\left(N - gram\right)}{\sum_{S \int Reference\ summaries} \sum_{N-gram\ \varepsilon\ S} Count\left(N - gram\right)} \tag{23}$$

where N N. iused for the length of the N-gram and Count $_{\text{match}}$ (N-gram) is the total number of N-grams co-occurring in a reference summary and a candidate summary. Count (N-gram) is the number of N-grams in the reference summaries. In our evaluation we used three metrics of ROUGE: ROUGE-1, ROUGE-2. We report the Recall score of ROUGE-1, ROUGE-2 to assess and compare our method, GDSCO, with other methods.

The crucial parameters that affect the performance of DE are as follows: the population size, $P = 150$; the number of iteration (fitness evaluation), $t_{\max} = 600$; the crossover rate, $CR = 0.7$. All of the results reported here are averaged over 30.

Performance Evaluation

In this section, the performance of our method is compared with other well-known or recently proposed methods. In particular, to evaluate our methods on DUC 2001and DUC 2002, we select the following methods: OMDPSO (Alguliev, Aliguliyev, & Mehdiyev, 2011), LexRank (Erkan & Radev, 2004), CollabSum (Wan, Yang, & Xiao, 2007), UnifieRank (Wan, 2010), 01-nonlinear (Alguliev, Aliguliyev, & Isazade, 2013b) and NetSum (Svore, Vanderwende, & Burges, 2007). These methods have been chosen for comparison because they have achieved the best results on the DUC2001and DUC2002 datasets. We also validate our proposed method, GDSCO, using a comparison of the overall ROUGE value obtained by GDSCO and the participating systems in DUC2001 and DUC2002. For this purpose, we selected the top three performing systems defined by DUC. The experimental results are reported in Tables 2 and 3. Table 2 shows the results of ROUGE metrics for GDSCO and other methods for the DUC2001 dataset. Table 3 displayed the experimental result s for DUC2002. In Table 2, System N, System P and System T are the top three performing systems for DUC2001. In Table 3, System19, System26, System28 are the top three performing systems for DUC2002. The top three systems are the systems with highest ROUGE scores.

We first run our method on the DUC2001 dataset, and then we extend the experiment on DUC2002 dataset with the same control parameters. Table 2, Table 3, Figure 1, and Figure 2 present the obtained results of ROUGE metrics. The obtained results prove that GDSCO outperforms the other examined methods and that our method produces very competitive results. The proposed GDSCO outperforms all the methods over all ROUGE metrics on the DUC2001 dataset. In addition, the ROUGE values of GDSCO are higher than that of the top three participating systems. On the DUC2002 dataset, GDSCO presents the best ROUGE values than that of the best participating systems and other methods.

Table 2. ROUGE scores of the methods on DUC2001

Methods	ROUGE-1	ROUGE-2
GDSCO	**0.4723**	**0.2103**
LexRank	0.4468	0.1989
CollabSum	0.4404	0.1623
OMDPSO	0.3993	0.0832
UnifiedRank	0.3636	0.0650
01-nonlinear	0.3876	0.0778
MA-DocSum	0.4486	0.2014
Netsum	0.4642	0.1769
System N	0.3391	0.0685
System P	0.3333	0.0665
System T	0.3303	0.0786

Detailed Comparison

With comparison to the ROUGE values for other methods, our method achieved significant improvement. Tables 4 and 5 show the improvement of GDSCO for all two ROUGE metrics. It is clear that GDSCO obtains the high ROUGE values and outperforms all the other methods. We use the relative improvement:

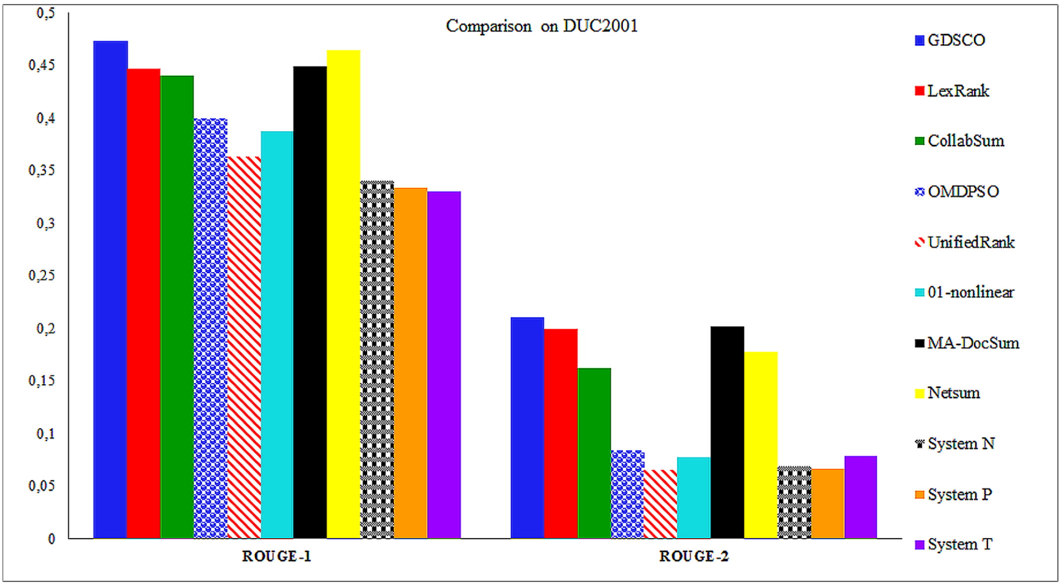
$$\left(\frac{Our\ \ method - Other\ \ method}{Other\ \ method} \times 100\right)$$

for comparison.

Table 3. ROUGE scores of the methods on DUC2002

Methods	ROUGE-1	ROUGE-2
GDSCO	**0.4901**	**0.2304**
LexRank	0.4796	0.2295
CollabSum	0.4719	0.2010
OMDPSO	0.4172	0.1026
UnifiedRank	0.3834	0.0786
01-nonlinear	0.4097	0.0937
MA-DocSum	0.4828	0.2284
Netsum	0.4496	0.1117
System 26	0.3515	0.0764
System 19	0.3450	0.0794
System 28	0.3436	0.0752

Figure 1. Comparison of the methods on DUC2001



In Tables 4 and 5 ‘‘+’’ means the proposed method improves the DUC systems and the related methods. Tables 4 and 5 present among other methods the MA-DocSum shows the best results compared to LexRank, CollabSum, OMDPSO, UnifiedRank, 01-nonlinear, Netsum, System N, System P, System T, System 19, System 26 and System 28. Compared with the method MA-DocSum on DUC2001 (DUC2002) dataset our method improves the performance by 5.28 (1.51%) and 4.42 (0.88%) in terms ROUGE-1 and ROUGE-2 metrics, respectively.

Figure 2. Comparison of the methods on DUC2002

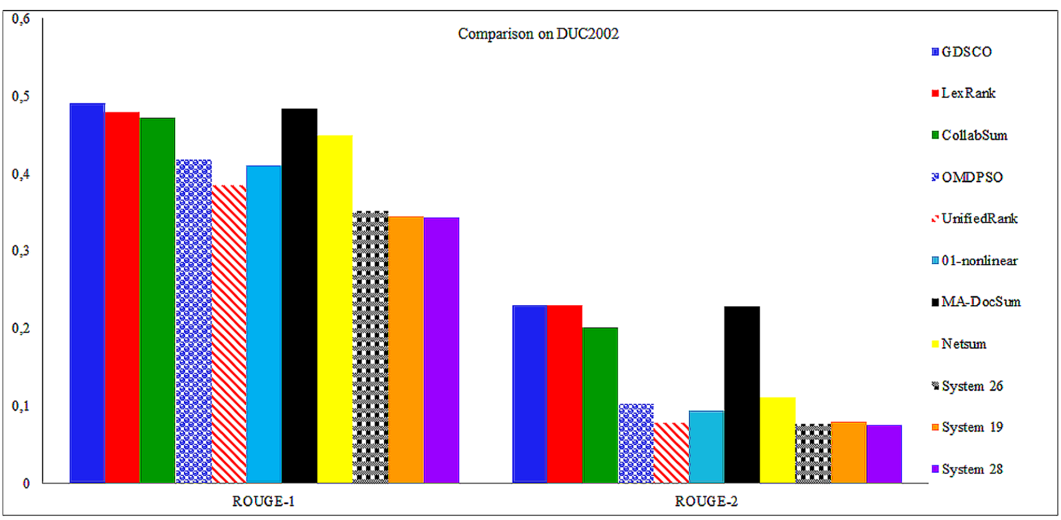


Table 4. Comparison of our method with other methods on the DUC2001 dataset

Methods	ROUGE-1	ROUGE-2
LexRank	+ 5.71	+ 5.73
CollabSum	+ 7.24	+ 29.57
OMDPSO	+ 18.28	+ 152.76
UnifiedRank	+ 29.90	+ 223.54
01-nonlinear	+ 21.85	+ 170.31
MA-DocSum	+ 5.28	+ 4.42
Netsum	+ 1.74	+ 18.88
System N	+ 39.28	+ 207.01
System P	+ 41.70	+ 216.24
System T	+ 42.99	+ 167.56

CONCLUSION

In this work, we proposed a method for generic document summarization. The method aims to optimize three properties a) coverage: summary should contain informative information that indicates the main idea of source text; b) diversity: summaries should not include the sentences that convey the same information; and c) length: summary is bounded in length. We select some important sentences from document(s) to produce a summary. We compared our methods with several existing summarization methods and the top several performing systems defined by DUC. We used DUC2001and DUC2001datasets to assess the performance of our method using the ROUGE metrics. The experimental results present that the proposed method is appropriate for the task of generic document summarization. The results also displayed that the GDSCO improved the performance of the participating system in DUC2001 and DUC2002; and the current methods. In future research, we also would like to extend our method with additional external knowledge such as semantic analysis.

Table 5. Comparison of our method with other methods on the DUC2002 dataset

Methods	ROUGE-1	ROUGE-2
LexRank	+ 2.19	+ 0.39
CollabSum	+ 3.86	+ 14.63
OMDPSO	+ 17.47	+ 124.56
UnifiedRank	+ 27.83	+ 193.13
01-nonlinear	+ 19.62	+ 145.89
MA-DocSum	+ 1.51	+ 0.88
Netsum	+ 9.01	+ 106.27
System N	+ 39.43	+ 201.57
System P	+ 42.06	+ 190.18
System T	+ 42.64	+ 206.38

REFERENCES

Alguliev, R. M., Aliguliyev, R. M., & Hajirahimova, M. S. (2012). Quadratic Boolean programming model and binary differential evolution algorithm for text summarization. *Problems of Information Technology*, *3*(2), 20–29.

Alguliev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2013a). CDDS: Constraint-driven document summarization models. *Expert Systems with Applications*, *40*(2), 458–465. doi:10.1016/j.eswa.2012.07.049

Alguliev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2013b). Formulation of document summarization as a 0–1 nonlinear programming problem. *Computers & Industrial Engineering*, *64*(1), 94–102. doi:10.1016/j.cie.2012.09.005

Alguliev, R. M., Aliguliyev, R. M., & Mehdiyev, C. A. (2011). An optimization model and DPSO-EDA for document summarization. *International Journal of Information Technology and Computer Science*, *3*(5), 59–68. doi:10.5815/ijitcs.2011.05.08

Alguliyev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2015). A new similarity measure and mathematical model for text summarization. *Problems of Information Technology*, *6*(1), 42–53.

Aliguliyev, R. M. (2010). Multidocument summarization through clustering and ranking of sentences[in Russian]. *Problems of Information Technology*, *1*(1), 26–37.

Bagirov, A. M., Ugon, J., & Webb, D. (2011). Fast modified global k-means algorithm for incremental cluster construction. *Pattern Recognition*, *44*(4), 866–876. doi:10.1016/j.patcog.2010.10.018

Balaji, J., Geetha, T. V., & Parthasarathi, R. (2014). A graph based query focused multi-document summarization. *International Journal of Intelligent Information Technologies*, *10*(1), 16–41. doi:10.4018/ijiit.2014010102

Cai, X., & Li, W. (2011). A spectral analysis approach to document summarization: Clustering and ranking sentences simultaneously. *Information Sciences*, *181*(18), 3816–3827. doi:10.1016/j.ins.2011.04.052

Cai, X., Li, W., & Zhang, R. (2013). Combining co-clustering with noise detection for theme-based summarization. ACM Transactions on Speech and Language Processing, 10(4).

Cai, X., Li, W., & Zhang, R. (2014). Enhancing diversity and coverage of document summaries through subspace clustering and clustering-based optimization. *Information Sciences*, *279*, 764–775. doi:10.1016/j.ins.2014.04.028

Canhasi, E., & Kononenko, I. (2014). Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Systems with Applications*, *41*(2), 535–543. doi:10.1016/j.eswa.2013.07.079

Cheng, J., Zhang, G., & Neri, F. (2013). Enhancing distributed differential evolution with multicultural migration for global numerical optimization. *Information Sciences*, *247*, 72–93. doi:10.1016/j.ins.2013.06.011

Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*(1), 4–31. doi:10.1109/TEVC.2010.2059031

Erkan, G., & Radev, D. (2004). LexRank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Ferreira, R., Cabral, L. S., Freitas, F., Lins, R. D., Silva, G. F., Simske, S. J., & Favaro, L. (2014). A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications*, *41*(13), 5780–5787. doi:10.1016/j.eswa.2014.03.023

Li, K., Kwong, S., & Deb, K. (2015). A dual-population paradigm for evolutionary multiobjective optimization. *Information Sciences*, *309*, 50–72. doi:10.1016/j.ins.2015.03.002

Lin, C.-Y. (2004). Rouge: a package for automatic evaluation of summaries.*Proceedings of the Workshop on Text Summarization Branches Out*, Barcelona, Spain (pp.74–81).

Lloret, E., & Palomar, M. (2013). COMPENDIUM: A text summarization tool for generating summaries of multiple purposes, domains, and genres. *Natural Language Engineering*, *19*(2), 147–186. doi:10.1017/S1351324912000198

Luo, W., Zhuang, F., He, Q., & Shi, Z. (2013). Exploiting relevance, coverage, and novelty for query-focused multi-document summarization. *Knowledge-Based Systems*, *46*, 33–42. doi:10.1016/j.knosys.2013.02.015

Mei, J.-P., & Chen, L. (2012). SumCR: A new subtopic-based extractive approach for text summarization. *Knowledge and Information Systems*, *31*(3), 527–545. doi:10.1007/s10115-011-0437-x

Nishino, M., Yasuda, N., Hirao, T., Suzuki, J., & Nagata, M. (2013). Text summarization while maximizing multiple objectives with Lagrangian relaxation.*Proceedings of the 35th European conference on Advances in Information Retrieval* (pp. 772-775). doi:10.1007/978-3-642-36973-5_81

Rakshit, P., & Konar, A. (2015). Extending multi-objective differential evolution for optimization in presence of noise. *Information Sciences*, *305*, 56–76. doi:10.1016/j.ins.2015.02.008

Song, W., Liang, J. Z., & Park, S. C. (2014). Fuzzy control GA with a novel hybrid semantic similarity strategy for text clustering. *Information Sciences*, *273*, 156–170. doi:10.1016/j.ins.2014.03.024

Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization*, *11*(4), 341–359. doi:10.1023/A:1008202821328

Svore, K. M., Vanderwende, L., & Burges, C. J. (2007). Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources.*Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (pp. 448–457).

Takamura, H., & Okumura, M. (2009). Text summarization model based on maximum coverage problem and its variant.*Proceedings of the 12th Conference of the European Chapter of the ACL*, Athens, Greece (pp. 781–789). doi:10.3115/1609067.1609154

Wan, X. (2010). Towards a unified approach to simultaneous single-document and multi-document summarizations.*Proceedings of the 23rd International Conference on Computational Linguistics: Association for Computational Linguistics* (pp. 1137–1145).

Wan, X., Yang, J., & Xiao, J. (2007). CollabSum: exploiting multiple document clustering for collaborative single document summarizations.*Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 143–150). doi:10.1145/1277741.1277768

Yang, C. C., & Wang, F. L. (2008). Hierarchical summarization of large documents. *Journal of the American Society for Information Science and Technology*, *59*(6), 887–902. doi:10.1002/asi.20781

Yang, L., Cai, X., Zhang, Y., & Shi, P. (2014). Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information Sciences*, *260*, 37–50. doi:10.1016/j.ins.2013.11.026

Rasim M. Alguliyev. He is director of the Institute of Information Technology of Azerbaijan National Academy of Sciences (ANAS) and academician-secretary of ANAS. He is full member of ANAS and full professor. He received BSc and MSc in electronic computing machines from the Azerbaijan Technical University in 1979. He received his PhD and Doctor of Science (higher degree after PhD) in Computer Science in 1995 and 2003, respectively. His research interests include: Information Security, Information Society, Online Social Network Analysis, Cloud Computing, Evolutionary and Swarm Optimization, Data Mining, Social Network Analysis, Big Data, and Scientometrics. He is author more than 520 papers, 4 monographs, 4 patents, several books.

Ramiz M. Aliguliyev. He is Head of Department at the Institute of Information Technology of ANAS. He received BSc and MSc in applied mathematics from the Baku State University, Azerbaijan in 1983. He received his Ph.D. (2002) in Mathematics and Doctor of Science (higher degree after PhD) in Computer Science (2011). His research interests include: Text Mining; Clustering; Evolutionary and Swarm Optimization; Web Mining; Online Social Network Analysis; Big Data Analytics and Scientometrics. He is author 45 papers and 3 books.

Nijat R. Isazade. He received BSc in computer science from the Baku State University, Azerbaijan in 2016. He is currently graduate student majoring in Software Systems Engineering at the RWTH Aachen University, Germany. His research interests include: Parallel Programming; Applied Statistics, Speech Recognition, Bioinformatics, Evolutionary and Swarm Optimization; and Big Data Analytics. He is author of 8 papers.

Asad Abdi received the M.Tech and Ph.D degrees in software engineering and computer science from Jawaharlal Nehru Technological University, India, 2011 and University of Malaya, Malaysia, 2016, respectively. From April 2015 to June 2016 he held a research assistant position at the University of Malaya. He is a member of the IEEE, UACEE and IACSIT. He published several papers in various ISI journals and international conference. He also is reviewer and member of editorial board of numerous journals. His research interests include Text mining, Natural Language processing (NLP), Question /Answering systems, Information Retrieval (IR), Plagiarism Detection, Text Summarization, Ontology and Semantic Network, Artificial Intelligence and Machine learning.

Norisma Idris received the Ph.D degrees in computer science from University of Malaya in 2011. She is currently Head of Department of Artificial Intelligence, University of Malaya. Her research interests include artificial Intelligence in Education (summarization, summary sentence decomposition, heuristic rules) and Natural Language (Malay text, text processing, stemming algorithm, automated essay grading system). She has published over 20 papers in various well-known conferences and journals. She is a co-author of a book.

How to Submit Proof Corrections Using Adobe Reader

Using Adobe Reader is the easiest way to submit your proposed amendments for your IGI Global proof. If you don't have Adobe Reader, you can download it for free at http://get.adobe.com/reader/. The comment functionality makes it simple for you, the contributor, to mark up the PDF. It also makes it simple for the IGI Global staff to understand exactly what you are requesting to ensure the most flawless end result possible.

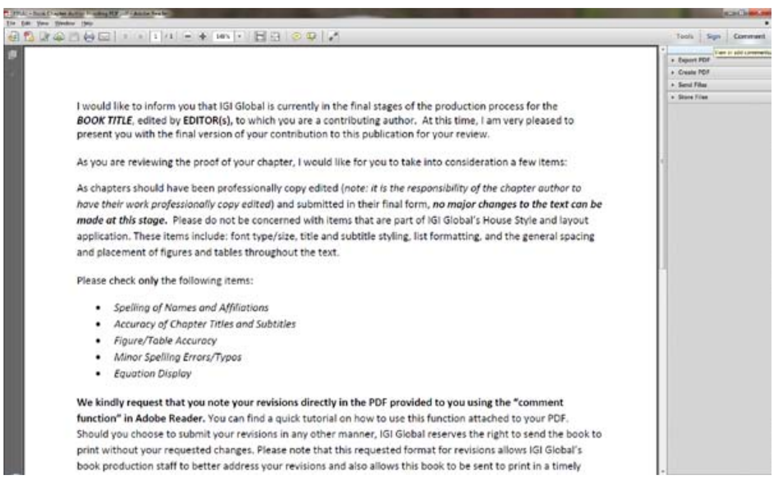
Please note, however, that at this point in the process the only things you should be checking for are:

Spelling of Names and Affiliations, Accuracy of Chapter Titles and Subtitles, Figure/Table Accuracy, Minor Spelling Errors/Typos, Equation Display

As chapters should have been professionally copy edited and submitted in their final form, please remember that ***no major changes to the text can be made at this stage*.**

Here is a quick step-by-step guide on using the comment functionality in Adobe Reader to submit your changes.

1. Select the **Comment** bar at the top of page to View or Add Comments. This will open the **Annotations** toolbar.



I would like to inform you that IGI Global is currently in the final stages of the production process for the *BOOK TITLE*, edited by **EDITOR(s)**, to which you are a contributing author. At this time, I am very pleased to present you with the final version of your contribution to this publication for your review.

As you are reviewing the proof of your chapter, I would like for you to take into consideration a few items:

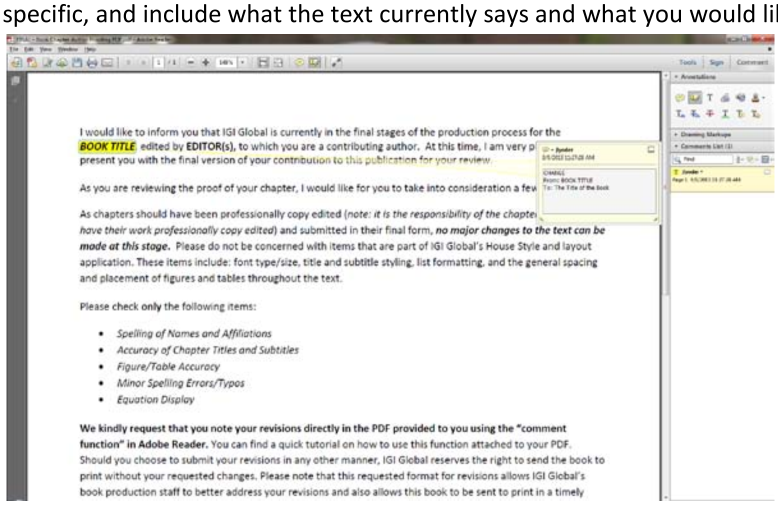
As chapters should have been professionally copy edited (*note: it is the responsibility of the chapter author to have their work professionally copy edited*) and submitted in their final form, ***no major changes to the text can be made at this stage.*** Please do not be concerned with items that are part of IGI Global's House Style and layout application. These items include: font type/size, title and subtitle styling, list formatting, and the general spacing and placement of figures and tables throughout the text.

Please check only the following items:

- *Spelling of Names and Affiliations*
- *Accuracy of Chapter Titles and Subtitles*
- *Figure/Table Accuracy*
- *Minor Spelling Errors/Typos*
- *Equation Display*

We kindly request that you note your revisions directly in the PDF provided to you using the "comment function" in Adobe Reader. You can find a quick tutorial on how to use this function attached to your PDF. Should you choose to submit your revisions in any other manner, IGI Global reserves the right to send the book to print without your requested changes. Please note that this requested format for revisions allows IGI Global's book production staff to better address your revisions and also allows this book to be sent to print in a timely

2. To note text that needs to be altered, like a subtitle or your affiliation, you may use the **Highlight Text** tool. Once the text is highlighted, right-click on the highlighted text and add your comment. Please be specific, and include what the text currently says and what you would like it to be changed to.



I would like to inform you that IGI Global is currently in the final stages of the production process for the *BOOK TITLE*, edited by **EDITOR(s)**, to which you are a contributing author. At this time, I am very pl

present you with the final version of your contribution to this publication for your review.

As you are reviewing the proof of your chapter, I would like for you to take into consideration a few

As chapters should have been professionally copy edited (*note: it is the responsibility of the chapte have their work professionally copy edited*) and submitted in their final form, ***no major changes to the text can be made at this stage.*** Please do not be concerned with items that are part of IGI Global's House Style and layout application. These items include: font type/size, title and subtitle styling, list formatting, and the general spacing and placement of figures and tables throughout the text.

Please check only the following items:

- *Spelling of Names and Affiliations*
- *Accuracy of Chapter Titles and Subtitles*
- *Figure/Table Accuracy*
- *Minor Spelling Errors/Typos*
- *Equation Display*

We kindly request that you note your revisions directly in the PDF provided to you using the "comment function" in Adobe Reader. You can find a quick tutorial on how to use this function attached to your PDF. Should you choose to submit your revisions in any other manner, IGI Global reserves the right to send the book to print without your requested changes. Please note that this requested format for revisions allows IGI Global's book production staff to better address your revisions and also allows this book to be sent to print in a timely