Alguliyev R.M.[1]***, Aliguliyev R.M.***[2]***, Isazade N.R.***[3]
Institute of Information Technology of ANAS, Baku, Azerbaijan
[1] r.alguliev@gmail.com; [2] r.aliguliyev@gmail.com; [3] nijatisazade@gmail.com

A NEW SIMILARITY MEASURE AND MATHEMATICAL MODEL FOR TEXT SUMMARIZATION

This paper proposes a new text similarity measure and mathematical model for automatic text summarization. Model consists of two stages. At the first stage, for detection of topics the sentences in document collection are clustered. At the second stage, the model generates a summary by extracting relevant sentences from each cluster. For clustering of sentences the k-means algorithm is utilized. Sentence selection process is formalized as an optimization problem. To select relevant sentences from each cluster and avoid redundancy in the summary this model uses both the sentence-to-cluster relation and the sentence-to-sentence relation. To solve the optimization problem a differential evolution algorithm with adaptive mutation strategy is developed.

Keywords: *new RRN similarity measure, sentence clustering, k-means, optimization model, differential evolution algorithm, modified mutation operator.*

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 network, 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 applications 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 [1, 2]. 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 [3, 4].

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 [1, 5]. *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 [1]. 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 [6–16].

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 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. The focus of the work [13] is to explore a systematic approach that allows diversity and coverage to be tackled within an integrated clustering-based summarization framework. Cai et al. [14] 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 [16], 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 [4] 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. [5] 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 [2], 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.

The use of optimization models for summarization purposes has also been investigated by many researchers. For example, in [17–26] 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) [20, 21], 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 [24] where the document summarization is modeled as a quadratic integer programming problem and solved with discrete binary particle swarm optimization algorithm. In [27], text summarization modeled as a maximum coverage problem that aims at covering as many conceptual units as possible by selecting some sentences.

Mathematical formulation of document summarization problem

Problem statement. Given a document collection $\mathbf{D}=\{D_1,...,D_N\}$, 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. $\mathbf{S}=\{S_1,...,S_n\}$, where S_i denotes i th sentence in $\mathbf{D}$, n is the number of sentences in the document collection. We attempt to find a subset of the sentences $\mathbf{S}=\{S_1,...,S_n\}$ 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 new RRN similarity measure. Let $\mathbf{T}=\{t_1,t_2,...,t_m\}$ represents all the distinct terms occurred in the document collection $\mathbf{D}$, 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, $S_i=[w_{i1},...,w_{im}]$, $i=1,...,n$, 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 [28] framework is utilized for indexing term weights:

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

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:

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

where n is the total number of sentences in the document collection $\mathbf{D}$; 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.

In text mining similarity measure plays an important role. Intuitively, if there are many common words between two sentences, they are very similar. Given two sentences $S_i = [w_{i1},...,w_{im}]$ and $S_j = [w_{j1},...,w_{jm}]$. To measure similarity between two sentences we introduce the following new *RRN* (*Rasim*, *Ramiz* & *Nijat*) measure:

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

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, $X = \{x_1, x_2,..., x_n\}$. The partition matrix may be represented as $U = [u_{iq}]$ ($i = 1,2,...,n$ and $q = 1,2,...,k$) where u_{iq} is the membership of pattern x_i to the q th 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 q th 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 $\mathbf{S} = \{s_1,..., s_n\}$ into a given number k of disjoint subsets C_q, $q = 1,2,...,k$, with respect to predefined criteria such that

1) for any $q = 1,2,...,k$ $C_q \neq \emptyset$, i.e. each cluster should have at least one sentence assigned;
2) for any $q1 \neq q2$ $C_{q1} \cap C_{q2} = \emptyset$, $q1, q2 = 1,2,...,k$, i.e. two different clusters should have no sentences in common;
3) $\bigcup_{q=1}^{k} C_q = \mathbf{S}$, i.e. each sentence should definitely be attached to a cluster;
4) no constraints are imposed on the clusters C_q, $q = 1,2,...,k$.

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

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 Eq.(8).

Step 2. Initialize the centers to k random locations in the collection $\mathbf{S}=\{S_1,...,S_n\}$ 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 $\mathbf{S}$, and cluster representatives O_q using the objective function [29]:

maximize

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

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 $u_{iq}=\begin{cases}1, \text{ if } S_i\in C_q\\ 0, \text{otherwise}\end{cases}$, $O_q=[w_1^q,...,w_m^q]$ is the center of cluster C_p, lth coordinate w_l^q of which is calculated as: $w_l^q=\frac{1}{|C_q|}\sum_{i=1}^{n} w_{il}u_{iq}$, $|C_q|$ is the number of sentences assigned to cluster C_q; $sim_{RRN}(S_i,O_q)$ is the similarity measure between $S_i=[w_{i1},...,w_{im}]$ and $O_q=[w_1^q,...,w_m^q]$.

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 [12]:

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

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

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. Sentence selection from each cluster we formulate as following optimization problem:

maximize

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

subject to

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

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

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 is the length of summary, l_i denotes the length of sentence S_i.

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

The objective function (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.

Adaptive differential evolution algorithm

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 [30], [31]. In our study, the optimization problem (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 [30, 31] 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, $U_p(t) = [u_{p,1}(t),...,u_{p,n}(t)]$, where $u_{p,s}(t)$ is the sth decision variable of the pth chromosome in the population, $s = 1,2,...,n$; $p = 1,2,...,P$.

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

$$u_{p,s}(0) = u_s^{\min} + (u_s^{\max} - u_s^{\min}) \cdot \text{rand}_{p,s}, \tag{12}$$

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

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 $U_p(t)$ from the same generation randomly chooses two other vectors $U_{p1}(t)$ and $U_{p2}(t)$, $p \neq p1 \neq p2$. Then it calculates the weighting combination of the U^{gbest}, the differences $(U_p^{lbest}(t) - U_p(t))$ and $(U_{p1}(t) - U_{p2}(t))$, and creates a trial offspring:

$$V_p(t) = U^{gbest}(t) + F(t) \cdot (U_p^{lbest}(t) - U_p(t)) + (1 - F(t)) \cdot (U_{p1}(t) - U_{p2}(t)) . \tag{13}$$

$U^{gbest}(t)$ 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:

$$F(t) = \frac{1}{1 + \exp(-t / t_{\max})}, \tag{14}$$

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

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 $Z_p(t) = [z_{p,1}(t),...,z_{p,n}(t)]$. It is developed from the elements of the target vector, $U_p(t)$, and the elements of the mutant vector, $Y_p(t)$:

$$z_{p,s}(t) = \begin{cases} v_{p,s}(t) \text{ if } \text{rand}_{p,s} \leq CR \text{ or } s = s^* \\ u_{p,s}(t) \text{ otherwise} \end{cases} . \qquad (15).$$

$CR \in [0,1]$ is the crossover constant which controls the recombination of target vector and mutant vector to generate trial vector and $s^* \in \{1,2,...,n\}$ 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. 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 values. 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. If the parent is still better, it is retained in the population:

$$U_p(t+1)=\begin{cases} Z_p(t), \text{ if } \ fit(Z_p(t)) \geq fit(U_p(t)) \\ U_p(t), \text{ otherwise} \end{cases}, \tag{16}$$

$fit(U)$ 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 [30, 31]: Then, the corresponding binary value of the sth element of the current target individual vector $U_p(t+1)$ is generated as Eq. (17) according to the probability $\text{rand}_{p,s}$ [32]:

$$u_{p,s}(t+1)=\begin{cases} 1, \text{ if } \ \text{rand}_{p,s} < sigm(u_{p,s}(t+1)) \\ 0, \text{ otherwise} \end{cases}, \tag{17}$$

where

$$sigm(z)=\frac{1}{1+\exp(-z)} \tag{18}$$

is the sigmoid function.

The motivation to use the sigmoid function is to map interval $[u_s^{\min}, u_s^{\max}]$ for each $s \in \{1,2,...,n\}$ 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 sth sentence is selected to be included to the summary, otherwise, the sth 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 $U_p(t+1) = [u_{p,1}(t+1),...,u_{p,n}(t+1)]$ to variables $\mathrm{X} = \{x_{iq}\}$ used for objective function calculation (9). This transformation can be written as follow:

$$x_{ip} = \begin{cases} 1, \text{ if } u_{p,i} = 1 \\ 0, \text{otherwise} \end{cases}. \tag{19}$$

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:

$$fit(X) = f(X) \cdot \exp(-\beta \cdot \max(0, \sum_{q=1}^{k} \sum_{i=1}^{n} l_i x_{iq} - L)), \tag{20}$$

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

The first multiplier $f(X)$ in Eq.(20) 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:

$$\beta = \beta^- + (\beta^+ - \beta^-) \frac{t}{t_{\max}},$$

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 $\beta^+ = 0.5$.

Conclusion

For effective multi-document summarization, it is important to reduce redundant information in the summaries and extract sentences that are common to given documents. This paper presents a document summarization model which extracts key sentences from given documents while reducing redundant information in summaries. The model is represented as a discrete optimization problem. To solve the discrete optimization problem we developed an adaptive DE algorithm.

References

1. Canhasi E., Kononenko I. Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization // Expert Systems with Applications, 2014, vol.41, no.2, pp.535–543.
2. Ferreira R., Cabral L.S., Freitas F., Lins R.D., Silva G.F., Simske S.J., Favaro L. A multi-document summarization system based on statistics and linguistic treatment // Expert Systems with Applications, 2014, vol.41, no.13, pp.5780–5787.
3. Yang C.C., Wang F.L. Hierarchical summarization of large documents // Journal of the American Society for Information Science and Technology, 2008, vol.59, no.6, pp.887–902.
4. Lloret E., Palomar M. COMPENDIUM: a text summarization tool for generating summaries of multiple purposes, domains, and genres // Natural Language Engineering, 2013, vol.19, no.2, pp.147–186.
5. Luo W., Zhuang F., He Q., Shi Z. Exploiting relevance, coverage, and novelty for query-focused multi-document summarization // Knowledge-Based Systems, 2013, vol.46, pp.33–42.
6. Alyguliyev R.M. The two-stage unsupervised approach to multi-document summarization // Automatic Control and Computer Sciences, 2009, vol.43, no.5, pp.276–284.
7. Aliguliyev R.M. Multidocument summarization through clustering and ranking of sentences // Problems of Information Technology, 2010, no.1, pp.26–37.
8. Aliguliyev R.M. 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, Hong Kong, China, December 18–22, 2006, pp.626–629.
9. Alguliev R.M., Alyguliev R.M. Summarization of text-based documents with a determination of latent topical sections and information-rich sentences // Automatic Control and Computer Sciences, 2007, vol.41, no.3, pp.132–140.
10. Alguliev R.M., Aliguliyev R.M. Automatic text documents summarization through sentences clustering // Journal of Automation and Information Sciences, 2008, vol.40, no.9, pp.53–63.
11. Aliguliyev R.M. Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization // Computational Intelligence, 2010, vol.26, no.4, pp.420–448.
12. Aliguliyev R.M. A new sentence similarity measure and sentence based extractive technique for automatic text summarization // Expert Systems with Applications, 2009, vol.36, no.4, pp.7764–7772.
13. Cai X., Li W., Zhan R. Enhancing diversity and coverage of document summaries through subspace clustering and clustering-based optimization // Information Sciences, 2014, vol.279, pp.764–775.
14. Cai X., Li W., Zhang R. Combining co-clustering with noise detection for theme-based summarization // ACM Transactions on Speech and Language Processing, 2013, vol.10, no.4, Article 16, 27 pages.
15. Yang L., Cai X., Zhang Y., Shi P. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization // Information Sciences, 2014, vol.260, pp.37–50.
16. Mei J.-P., Chen L. SumCR: A new subtopic-based extractive approach for text summarization // Knowledge and Information Systems, 2012, vol.31, no.3, pp.527–545.
17. Alguliev R.M., Aliguliyev R.M., Hajirahimova M.S., Mehdiyev C.A. MCMR: maximum coverage and minimum redundant text summarization model // Expert Systems with Applications, 2011, vol.38, no.12, pp.14514–14522.

18. Alguliev R.M., Aliguliyev R.M., Mehdiyev C.A. An optimization model and DPSO-EDA for document summarization // International Journal of Information Technology and Computer Science, 2011, vol.3, no.5, pp.59–68.
19. Alguliev R.M., Aliguliyev R.M., Mehdiyev C.A. Sentence selection for generic document summarization using an adaptive differential evolution algorithm // Swarm and Evolutionary Computation, 2011, vol.1, no.4, pp.213–222.
20. Alguliev R.M., Aliguliyev R.M., Hajirahimova M.S. GenDocSum + MCLR: Generic document summarization based on maximum coverage and less redundancy // Expert Systems with Applications, 2012, vol.39, no.16, pp.12460–12473.
21. Alguliev R.M., Aliguliyev R.M., Hajirahimova M.S. Quadratic Boolean programming model and binary differential evolution algorithm for text summarization // Problems of Information Technology, 2012, no.2, pp.20–29.
22. Alguliev R.M., Aliguliyev R.M., Isazade N.R. DESAMC+DocSum: differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization // Knowledge-Based Systems, 2012, vol.36, pp.21–38.
23. Alguliev R.M., Aliguliyev R.M., Mehdiyev C.A. An optimization approach to automatic generic document summarization // Computational Intelligence, 2013, vol.29, no.1, pp.129–155.
24. Alguliev R.M., Aliguliyev R.M., Isazade N.R. CDDS: Constraint-driven document summarization models // Expert Systems with Applications, 2013, vol.40, no.2, pp.458–465.
25. Alguliev R.M., Aliguliyev R.M., Isazade N.R. Multiple documents summarization based on evolutionary optimization algorithm // Expert Systems with Applications, 2013, vol.40, no.5, pp.1675–1689.
26. Alguliev R.M., Aliguliyev R.M., Isazade N.R. Formulation of document summarization as a 0‑1 nonlinear programming problem // Computers and Industrial Engineering, 2013, vol.64, no.1, pp.94–102.
27. Takamura H., Okumura M. 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, March 30 - April 3, 2009, pp.781–789.
28. Song W., Liang J.Z., Park S.C. Fuzzy control GA with a novel hybrid semantic similarity strategy for text clustering // Information Sciences, 2014, vol.273, pp.156–170.
29. Bagirov A.M., Ugon J., Webb D. Fast modified global k-means algorithm for incremental cluster construction // Pattern Recognition, 2011, vol.44, no.4, pp.866–876.
30. Storn R., Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous space // Journal of Global Optimization, 1997, vol.11, no.4, pp.341–359.
31. Das S., Suganthan P.N. Differential evolution: a survey of the state-of-the-art // IEEE Transactions on Evolutionary Computation, 2011, vol.15, no.1, pp.4–31.
32. Alguliev R.M., Aliguliyev R.M. Evolutionary algorithm for extractive text summarization // Intelligent Information Management, 2009, vol.1, no.2, pp.128–138.

UOT 004.9

Əliqulyev Rasim M.[1]**, Alıquliyev Ramiz M.**[2]**, İsazadə Nicat R.**[3]
AMEA İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan
[1] r.alguliev@gmail.com; [2] r.aliguliyev@gmail.com; [3] nijatisazade@gmail.com

Mətnlərin referatlaşdırılması üçün yeni yaxınlıq ölçüsü və riyazi model

Məqalədə mətnlərin referatlaşdırılması üçün yeni yaxınlıq ölçüsü və riyazi model təklif edilmişdir. Model iki mərhələdən ibarətdir. Birinci mərhələdə, tematik bölmələri aşkarlamaq üçün sənədlər çoxluğundakı cümlələr klasterləşdirilir. İkinci mərhələdə, hər bir klasterdən cümlələri seçməklə referat yaradılır. Cümlələri klasterləşdirmək üçün k-means alqoritmi istifadə olunmuşdur. Klasterlərdən cümlələrin seçilməsi optimallaşdırma məsələsi kimi formalizə edilmişdir. Klasterlərdən relevant cümlələrin seçilməsi və məzmuna görə yaxın cümlələrin referatda iştirakını minimallaşdırmaq üçün cümlələrlə klasterlər, o cümlədən cümlələrin öz aralarındakı semantik yaxınlıq nəzərə alınmışdır. Optimallaşdırma məsələsinin həlli üçün adaptiv mutasiya strategiyasına malik diferensial evolyusiya alqoritmi işlənmişdir.

Açar sözlər: *yeni RRN yaxınlıq ölçüsü, cümlələrin klasterləşdirilməsi, k-means, optimallaşdırma modeli, diferensial evolyusiya alqoritmi, modifikasiya olunmuş mutasiya operatoru.*

УДК 004.9

Алгулиев Расим М.[1]**, Алыгулиев Рамиз М.**[2]**, Исазаде Ниджат Р.**[3]
Институт Информационных Технологий НАНА, Баку, Азербайджан
[1] r.alguliev@gmail.com; [2] r.aliguliyev@gmail.com; [3] nijatisazade@gmail.com

Новая мера подобия и математическая модель для реферирования текстов

В данной работе предлагаются новая мера подобия и математическая модель для автоматического реферирования текстов. Модель содержит два этапа. На первом этапе для выявления тем предложения в наборе документов кластеризованы. На втором этапе модель с выбором релевантных предложений из каждого кластера формирует реферат. Для кластеризации предложений использован алгоритм *к*-средних. Процесс выбора предложений формализован как задача оптимизации. Для выбора релевантных предложений из каждого кластера и избегания дублирования предложений, близких по контенту, при выборе использованы как семантическое отношение между предложением и кластером, так и семантическое отношение между предложениями. Для решения задачи оптимизации разработан алгоритм дифференциальной эволюции с адаптивной стратегией мутации.

Ключевые слова: *новая мера близости RRN, кластеризация предложений, k-средних, оптимизационная модель, алгоритм дифференциальной эволюции, модифицированный оператор мутации.*