Journal of Computer & Robotics 15(1), 2022 9-17

Computer
& Robotics

A Method for Multi-text Summarization Based on Multi-Objective Optimization use Imperialist Competitive Algorithm

Amir Shahab Shahabi[a], Mohammad Reza Kangavari[b,*], Amir Masoud Rahmani[c]

[a]Department of Computer Engineering, Science and Research Branch, Islamic Azad University,Tehran, Iran

[b]Department of Computer Engineering, Science and Industry University,Tehran, Iran

[c]Department of Computer Engineering, Science and Research Branch, Islamic Azad University,Tehran, Iran

Received 17 December 2021; Accepted 17 March 2022

Abstract

In this research, we discuss the methods that have been proposed so far to solve automatic summarization, in which both single-text and multi-text are summarized with emphasis on experimental methods and text extraction techniques. In multi-text summarization, retrieving redundant information that is readable and coherent and contains maximum information from the original text and minimum redundancy has made research in this field very important. An extraction approach based on several methods for identifying sentence similarities and a meta-heuristic optimization algorithm that has been modified and optimized for faster convergence is presented. In this algorithm, changes are made based on density detection through the probability distribution function to avoid being placed in local optimization and try to search more extensively for the response space. The experimental results obtained from the implementation of the algorithm show that the efficiency on criteria such as ROUGE and the accuracy of the proposed method is effectively increased.

Keywords: Automatic Summarization, Optimization, ICA Algorithm, Single-text, Multi-text

1. Introduction

Today, information extraction systems are particularly important due to the increase in the volume of information on various topics. However, it is more important to have a system that can provide the user with an abstract or summary of the recovered data set. Much work has been done in this area [1].

In [2], the developed Bayesian method for thematic clustering of Persian texts based on data with an instructor was presented. Moreover, the study examined the belonging of a phrase to a cluster based on a probability function. The research was done on Persian words and phrases.

Bahorshpour developed a method for summarizing the Persian language based on the diagram of genetic algorithms [3]. In the proposed summary system, an attempt has been made to eliminate the shortcomings of the existing systems. The proposed system of summarizing Persian texts produces an extracted summary. The idea used in this summarizer is a combination of graph-based methods and genetic algorithms. This system produces a direction chart after measuring the sentences and forming a similarity matrix for the document sentences. The authors considered the initial population of the genetic algorithm to be the total number of sentences in the text. If the sentence is summarized, the chromosome value is one; and if the sentence is not in summary, the chromosome value is zero. We are to optimize the solution as much as possible to proceed with the Persian language summarization system. In another research, the automatic Text Summarization problem is formulated as a multi-objective optimization problem, and to mitigate this problem, the modified cat swarm optimization (MCSO) strategy is employed. In this work, the

* Corresponding Author. Email: kangavari@iust.ac.ir

population is represented as a collection of feasible individuals where the summary length limit is considered a constraint that determines the feasibility of an individual. Each individual is shaped by randomly selecting some of the sentences encoded in binary form.

Furthermore, two objective functions, namely “coverage and informativeness” and “anti-redundancy”, are used to evaluate each individual’s fitness. Also, to update the position of an individual, genetic and bit manipulating operators and the best cat memory pool have been incorporated into the system. Finally, from the generated non-dominated optimal solutions, the best solution is selected based on the ROUGE score for the summary generation process [9]. In the other research, the authors avoided engaging unsolvable text extracting when facing huge documents; they used meta-heuristic techniques. They used Cuckoo Search Optimization Algorithm (CSOA) to improve the performance of the extractive-based summarization method. The proposed approach is examined on Doc. 2002 standard documents and analyzed by Rouge evaluation software. The obtained results indicate better performance of the proposed method compared with other similar techniques [10].
Another research proposes a Multi-Objective Artificial Bee Colony Algorithm based on Decomposition (MOABC/D) to solve the extractive multi-document text summarization problem. An asynchronous parallel design of the MOABC/D algorithm has been implemented to take advantage of multi-core architectures. Experiments have been carried out with Document Understanding Conferences (DUC) datasets, and the results have been evaluated with Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics [11]. Another research proposed an automatic, generic, and extractive Arabic multi-document summarization system. The proposed system employed clustering-based and evolutionary multi-objective optimization methods. The clustering-based method discovered the main topics in the text, while the evolutionary multi-objective optimization method optimized three objectives based on coverage, diversity/redundancy, and relevancy. The performance of the proposed system is evaluated using TAC 2011 and DUC 2002 datasets [12].

In another paper, the authors introduced and formally defined the concepts of supplementary and complementary multi-modal summaries in the context of the overlap of information covered by different modalities in the summary output. A new problem statement of combined complementary and supplementary multi-modal summarization (CCS-MMS) is formulated. The problem is then solved in several steps by utilizing the concepts of multi-objective optimization by devising a novel unsupervised framework. An existing multi-modal summarization data set is further extended by adding outputs in different modalities to establish the efficacy of the proposed technique [13]. In another research Spider Monkey Optimization (SMO) algorithm is introduced for a summary generation. Before that, multi-documents are compressed into a single document, and different pre-processing methods are applied to remove the unwanted word from the document. Then, semantic and syntactic features are extracted from the document using different methods. The mined features are then provided into the softmax regression (SR) technique for further processing. Finally, the SMO algorithm is proposed to generate a summary of the whole document. The proposed text summarization process is implemented in the Python platform using the BBC news dataset, DUC (Document Understanding Conference) 2002, 2006, and 2007 datasets [14].

2. Criteria and Factors of a Good Summary

Here are five criteria for a good summary, which are as follows: Topic Relevance(TRF)
A good summary includes sentences similar to the topic of the documents. We can say that the average similarity of the sentences in summary divided by the maximum averages in all the summaries that can be produced is a factor called the topic relevance factor [15], which gives the following relation:

$$TRF_s = \frac{TR}{\max_{\forall\, summary}(TR)} \tag{1}$$

$$TR\_s = (\sum\_(s\_j \in \ summary) sim(S\_i, q)\ )/s \tag{2}$$

Where q is the topic, and the maximum on all summaries is calculated with length S. If this factor is close to 1, the summary produced is close to the document topic.
Cohesion(CF)
Cohesion Factor or (CF) indicates whether the sentences, in summary, speak about a matter or not. A good summary includes sentences that are perfectly related. We need the similarity of each pair

of sentences in the text to achieve this factor. It can be achieved using the neighborhood matrix of the text graph. So:

$$N_s = (S-1) + (S-2) + \cdots = \frac{(S) \times (S-1)}{2} \quad (3)$$

$$C_s = \frac{\sum_{\forall\, s_i, s_j \in\, summarysulgraf} W(s_i, s_j)}{N_s} \quad (4)$$

C in the above phrase is the average similarity of all the sentences in summary, which is the average weight of all the edges below the summary graph. N_s is the total number of edges below the summary graph, which can be easily calculated. So that if S is the total number of sentences, in summary, we add the length of a sequence of edges containing two, three, and S sentences together, which will be as shown above. With these characteristics, the cohesion factor can be calculated as follows:

$$CF_s = \frac{\log(C \times 9 + 1)}{\log(M \times 9 + 1)} \quad (5)$$

$$M = \max_{i,j \le N} sim_{i,j} \quad (6)$$

M is the maximum weight in the graph or the maximum similarities between the sentences. It is clear thatC≤M so that we will have:

$$\begin{gathered} 0 \le C \le M \;\Rightarrow 0 \le CF \le 1 \\ 0 \le C \le 1 \;\Rightarrow 1 \le 9 \times C + 1 \le 10 \Rightarrow 0 \le \log(9 \times C + 1) \le 1 \\ Similarly, 0 \le \log(9 \times M + 1) \le 1 \end{gathered} \quad (7)$$

After a few experiments with this formula using the base ten logarithm, this helps with cases where the mean is much smaller than the maximum weights or similarities, and by converting the division to a logarithmic domain, very small CF sizes will be prevented. If the subject of the sentences, in summary, focuses on one matter, the value of this factor will increase.

Readability(RF)

The readability factor (RF) in a text is defined as follows: a readable document contains a text in which a sentence is well related to its previous sentence, or in other words, the following statement is maximized:

$$sim_{s_s 1, s_s 2} + sim_{s_s 2, s_s 3} + \cdots + sim_{s_{s-1}, s_{s_s}} \quad (8)$$

In other words, a readable summary consists of sentences that form a soft chain. So we define the readability of the summary s by length S of the sentence as follows:

$$R_s = \sum_{0 \le i \le S} W(s_i, s_{i+1}) \quad (9)$$

$$RF_s = \frac{R_s}{\max_{\forall i} R_i} \quad (10)$$

The higher the value, the more readable the text will be. In the following section, we will discuss not repeating the matter in summary.

Anti-Redundancy (ARF)

Anti-Redundancy Factor (ARF) is a factor that prevents duplicate sentences from being summarized and makes the content fresh. Because the research of this thesis is related to multi-text or multi-document summarization, the compactness factor discussed earlier, and anti-redundancy or absence of repetitive sentences is very important. For this purpose, we use the relation provided in [16] and express it by applying changes as follows. In this regard, the novelty of the sentences among the existing documents while their relationship to the subject of the text is examined, and by the linear combination of these two factors, their repetition, in summary, is prevented.

We define this factor as follows:

$$Nonelty_s = N_s = \sum_{s_i \in R \backslash S} \sum_{s_j \in S} sim(s_i, s_j) \quad (11)$$

$$Relevance_s = R_s = \sum_{s_j \in R \backslash S} sim(s_i, q) \quad (12)$$

Where S is the set of sentences selected and placed in the summary, R is the set of sentences inside all documents, and R/S is the set of sentences not selected to be included in the summary. The factor that causes non-repetition while being similar to the subject or topic of the texts arises from the linear combination of the two factors of novelty and relevance. This is why it is called the maximum marginal relevance or MMR and is given as follows:

$$MMR_s = \frac{\lambda R_s + (1-\lambda) N_s}{|R| - |S|} \quad (13)$$

$$MMRF_s = \frac{MMR}{\max_{\forall summary}(MMR)} \quad (14)$$

$$0 \le \lambda \le 1$$

In creating the criterion for the above factors, the compactness of the generated summary is assumed so that in all cases, the number of summary sentences is considered constant and equal to S, while the total number of sentences in all texts is N.

3. Rouge

A set of criteria was produced and named ROUGE, which became a standard for automatically evaluating summaries [6].
Suppose $R = \{r_1, \ldots, r_m\}$ is a collection of reference summaries, and s is a summary generated by the system automatically. Again, suppose $\phi_n(d)$ is a binary vector representing the existence or presence of n-grams in text d. In this case, the *i*th component of$\phi_n^i(d)$, if there are the *i*th n-grams in d, it would be 1. Otherwise, it would be zero. Criterion ROUGE-N is an n-gram recall-based statistic that can be calculated:

$$ROUGE - N(s) = \frac{\sum_{r\in R}\langle \phi_n(r), \phi_n(s)\rangle}{\sum_{r\in R}\langle \phi_n(r), \phi_n(r)\rangle} \quad (15)$$

Where $\langle 0.0 \rangle$ is the common internal multiplication of the vectors, this criterion is very close to the BLUE criterion, based on accuracy. Unlike other criteria seen before, ROUGE-N can be used for multi-reference summaries, which is quite useful in practical situations. Another suggestion provides the most similar summary in the reference set:

$$ROUGE - N_{multi}(s) = \max_{r\in R} \frac{\langle \phi_n(r), \phi_n(s)\rangle}{\langle \phi_n(r), \phi_n(r)\rangle} \quad (16)$$

4. Objective Function

One of the easiest ways to deal with multi-objective issues is to accumulate the weight of its goals. A weight problem is first assigned to each objective function and multiplied in this method. Then the sum of all objective functions multiplied by their weight percentage forms a single-objective objective function that can be optimized by algorithms used to optimize single-objective problems. In the case of our project, this weight accumulation function can be defined as follows:

$$f_s = \frac{\alpha \times TRF_s + \beta \times CF_s + \gamma \times RF_s + \delta \times MMRF_s}{\alpha + \beta + \gamma + \delta} \quad (17)$$

The coefficients of each criterion can be adjusted according to its importance in summary. Therefore, it is better to define the objectives in the following way as follows as four components of a vector, which we call the target vector, and then, based on what is explained later, by examining the dominance of the target vectors in the target space over each other, find the dominant vectors and if the vectors could not overcome each other, use a second method to determine their superiority called crowding distance to recognize their superiority over each other and select the better target vector and from it obtain the optimal points in the solution space, which are the same sentences suitable for being summarized [7]; therefore, the objectives are defined as follows:

$$\begin{aligned} f_1 &= TRF_s \\ f_2 &= CF_s \\ f_3 &= RF_s \\ f_4 &= MMRF_s \end{aligned} \quad (18)$$

Moreover, the target vector is as follows:

$$f_s = (f_1, f_2, f_3, f_4) \quad (19)$$

5. Weighting Words and Sentences

Weighting words and sentences and finally, the edges of the graph is done using a combination of three methods:

- Use of tf/isf factor
- Use of Normalized Google Distance (NGD) factor
- Use of the fuzzy similarity method

5.1. Use of tf/isf Factor

This factor is calculated as follows:

$$tf_{i,j} = \frac{freq_{i,j}}{\max_l freq_{i,j}} \qquad isf_i = \log\frac{N}{n_i} \quad (20)$$

$tf_{i,j}$ indicates the frequency of the ith phrase in the jth sentence, and isf_i is the sentence deficiency of the ith phrase, which N is the total number of sentences and n_i is the number of sentences containing the ith phrase.

5.2. The first Case: Cosine Similarity of Sentences

So the cosine similarity between the vectors of two sentences will be as follows:

$$sim1(s_m, s_n) = \frac{\overrightarrow{d_m} \cdot \overrightarrow{d_n}}{|\overrightarrow{d_m}| \times |\overrightarrow{d_n}|} = \frac{\sum_{i=1}^{t} w_{i,m} \times w_{i,n}}{\sqrt{\sum_{i=1}^{t} w_{i,m}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,n}^2}} \quad (21)$$

Where t is the number of phrases in the sentences, according to this relation, the closer the similarity of two sentences in terms of the number of repetitions of words or phrases in them, the smaller the angle between the two vectors will be and vice versa.

5.3. The Second Case: Similarity of Normalized Google Distance of Sentences

Normalized Google Distance or NGD of sentences is used in the information retrieval system of Google site to summarize the findings [8]. So that the similarity of NGD between phrases t_k and t_l is defined as follows:

$$NGD(t_k, t_l) = \frac{\max\{\log(f_k), \log(f_l)\} - \log(f_{kl})}{\log n - \min\{\log(f_k), \log(f_l)\}} \quad (22)$$

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

Accordingly, the similarity between two sentences is defined by the following equation:

$$sim2(S_i, S_j) = \frac{\sum_{t_k \in S_i} \sum_{t_l \in S_j} sim_{NGD}(t_k, t_l)}{m_i m_j} \quad (23)$$

$$sim2(S_i, S_j) = \frac{\sum_{t_k \in S_i} \sum_{t_l \in S_j} sim_{NGD}(t_k, t_l)}{m_i m_j} \quad (23)$$

5.4. Third Case: Fuzzy Similarity between Sentences

In this method, first, the fuzzy knowledge base of Persian words must be formed [10]. All Persian words are placed in a fuzzy relation. If two words have no semantic relation with each other, their belonging function becomes zero; otherwise, it can take a value of up to 1 based on the amount of semantic relation. This fuzzy relation is defined as follows:

$$\widetilde{P} = \{((w_1, w_2), \mu_{\widetilde{P}}(w_1, w_2)) | (w_1 \times w_2) \in (W \times W)\} \quad (24)$$

Which is a relation from W, a set of words on itself. We create a fuzzy link with the number of components equal to the number of words and name it for each sentence. $\widetilde{R_i}$ In this sentence, if there is a word in Persian, the value of its membership function in relation will be 1. Otherwise, it will be zero. Now we combine the relation of this sentence with the relation of Persian words as fuzzy. This combination is called Fuzzy Max-Min Composition and is displayed as $\widetilde{R}_i o \widetilde{P}$. In this regard, similar words with a degree of belonging less than one and words that are exactly in the sentence are inserted with a degree of belonging 1. Now the similarity of the sentences can be determined based on the fuzzy similarity relation, which is done as follows:

$$Sim3(S_i, S_j) = \frac{|R_{y_i} \cap R_{y_j}|}{\min\{|R_{y_i}|, |R_{y_j}|\}} \quad (25)$$

5.5. The similarity of Two Sentences based on Combined Method

We calculate the similarity of the sentences based on the weight composition of the methods mentioned above. Since the fuzzy similarity method has a semantic origin and the previous two methods have a statistical origin, we consider the weight of the third method more, but these weights can be set and adjusted by the user as a parameter, which we will analyze in the next chapter by determining its values. The similarity relationship is as follows:

$$sim(S_i, S_j) = \frac{\alpha.sim1(S_i,S_j) + \beta.sim2(S_i,S_j) + \gamma.sim3(S_i,S_j)}{\alpha+\beta+\gamma} \quad (26)$$

6.Multi-objective Modified Imperialist Competitive Algorithm for Use in This Issue

The original Imperial Competitive Algorithm (ICA) in this issue has become a Multi-objective modified Imperial Competitive Algorithm (MOMICA). First, we describe the method of the initial algorithm, and then we explain the modifications and changes applied to make it multi-objective. In this algorithm, the following steps are performed:

- Creating initial answers and shaping primitive empires and colonies
- Defining objective functions
- Defining attraction or assimilation policy
- Combining the classical optimization method to increase the convergence speed
- Determining the Pareto front by ranking and determining the congestion distance of objectives and selecting the more appropriate set of objectives
- Calculating the power of the empire and, if necessary, swapping the colony for the empire
- The colonial rivalry between empires
- Fall of a weak empire
- Convergence

For simplicity, vector components can only be considered as the value of the membership function of a sentence with a summary. For simplicity, vector components can only be considered as the value of the membership function of a sentence with a summary. In order to quantify the membership function of each sentence, in summary, we use statistical methods that give points to the sentences in the text and are used in ordinary summarizer systems. These points are the initial value and may change during the algorithm to achieve the desired summary. The points include the following, listed in various papers and integrated in [17].
Sentence position: A maximum of the first five sentences of the paragraph are of this importance and are scored according to the position of the sentence as follows:

$$f1(s_i) = \frac{5 - sentence - position + 1}{5} \quad (27)$$

The similarity of Sentence with Topic: If a sentence is more similar to the topic q based on the similarity relation mentioned above, it will get more points. In the following equation, the sigmoid function makes the resulting number between 0 and 1.

$$f2(s_i) = \frac{1}{1 + e^{-sim(s_i,q)}}, \forall i = 1, N \quad (28)$$

Presence of a Proper Noun in a Sentence: Usually, sentences that contain several proper nouns are most important, so having a database of these nouns, we can consider a point for a sentence:

$$f3(s_i) = \frac{\#\,(propernames(s_i))}{Length(s_i)} \quad (29)$$

Existence of Numerical Information in a Sentence: Usually, sentences that contain numerical information can be more important to be included in the summary so that they can be graded:

$$f4(s_i) = \frac{\#\,(numericaldata(s_i))}{Length(s_i)} \quad (30)$$

Relative Sentence Length: This feature prevents short sentences from being selected for inclusion in summary.

$$f5(s_i) = \frac{Length(s_i)}{\mu_{Length(s_i)} + 3\sigma_{Length(s_i)}} \quad (31)$$

Dense Path of Nodes: The density of the node or business in the graph of sentences is defined by the number of connections of the node, which represents a sentence in the graph, to the other nodes, and as mentioned earlier, the weight of the graph edges that make up the same connections is the similarity between the nodes or the same sentences. This criterion can be calculated as follows for sentences:

$$f6(s_i) = \frac{\#\,(branches\_connected\_to\_the\_node\_of\_s_i)}{\#\,(branches\_of\_highest\_bushy\_node)} \quad (32)$$

Cumulative Similarity of Each Node (Sentence): By obtaining the sum of the weights of the edges connected to a node, this criterion expresses the semantic importance of that sentence concerning other sentences. By dividing this value by the maximum sum for a node with the highest weight, its path can be normalized. That is, as follows:

$$f7(s_i) = \frac{\sum S_i\_Branches\_Weights}{\sum Longest\_Node\_Branches\_Weights} \quad (33)$$

Statistical Score of Sentence: The more important sentences can be identified with the modified criterion based on the entropy of the most frequent phrase in blocks and texts:

$$W(T_i) = 1 + \log\left(\frac{tf_{ir}/f_i}{M}\right) \times tf_{ir}/f_i \tag{34}$$

tf_ir Number of times the phrase T_i is repeated in block r, and f_i is the total number of repetitions of the phrase T_i in the whole document or documents. Also, M is the number of blocks in the document or documents.

Now, if there are phrases with synonymous words in the whole text or texts, we add their weight to the weight of the original phrase according to the block in which they are as follows:

$$\begin{aligned} W(T_i) &= \sum_{T_j \in Document, Relation\ (T_j,T_i)=synonym} W(T_j) \\ &+ 0.7 \times \sum_{T_i,T_j \in B_k, Relation\ (T_j,T_i) \neq synonym} W(T_j) + 0.5 \\ &\times \sum_{T_i \in B_k, T_j \in B_m, k \neq m, Relation(T_j,T_i) \neq synonym} W(T_j) \end{aligned} \tag{35}$$

Now you can give weight to each sentence of the block so that the weight of each sentence will be the sum of the points of the words or phrases divided by the total number of words or phrases in that sentence.

$$f8(s_i) = \frac{\sum_{T_i \in s_i} W(T_i)}{Num(T_i)} \tag{36}$$

Point of Collocation of Phrases in Sentences: In this section, sentences are scored based on phrases and words that come together in different parts of the text. The degree of collocation of related phrases is calculated as follows:

$$R(w_i|w_j) = \frac{f(w_i,w_j)}{f(w_j)} \tag{37}$$

Where f(w_j) is the number of repetitions of the word w_j, and f(w_i,w_j) is the number of times the two phrases or words come together in a range of text. The size of this range is important, and it is better to consider the average length of the sentences. So, in general, the average degree of collocation is as follows:

$$C(w_i,w_j) = \frac{\left(R(w_i|w_j) + R(w_j|w_i)\right)}{2} \tag{38}$$

Based on this relation, the degree of collocation for both words is calculated, and then the lexical chain is made based on the information of synonymous words. Due to ambiguity in the meaning of words, a word can belong to one or more chains. In order to select the correct chain for each new word, the sum of the degree of collocation of each member of all chains is calculated, and the chain with the highest value becomes a candidate to insert the word in that chain. A new chain is created for that word if no chain is found.

$$\begin{aligned} & w_i \in Chain_d where: \\ & d = \max\{\sum_{j=1,lexical\_chain\_count} \sum_{W_k \in Chain_j} C(w_i,w_k)\} \\ & Important_{ce(w_i)} = Frequency(w_i) \\ & \qquad \times Chainmemebercount(d_{wi}) \end{aligned} \tag{39}$$

We rank each word by multiplying the repetition of a word in the number of chains it belongs to. Now, by clustering the text based on the degree of collocation, the points of the sentences can be obtained. At first, the probability that a word belongs to a cluster is equal to:

$$p(w \in C_i) = \frac{\sum_{w_c \in C_i, w \neq w_c} C(w,w_c)}{\sum_{w_c \in C_i, w \in D, w \neq w_c} \sum C(w_c,w)} \tag{40}$$

The numerator expresses the dependence of the word on a cluster, and the denominator expresses the dependence of the word on all clusters, so the probability that a word belongs to more than one cluster is as follows. Value of Count is the number of clusters.

$$LinkScore(w) = 1 - \sum_{i=1}^{Count} p(w \in C_i) \times p_{i \neq j}(w \notin C_j) \tag{41}$$

Now we select n words with the highest "Link Score" and add them to the graph. The value of each word is calculated by adding the weight of all the edges attached to that word. To get the point of each sentence, we can add the value of its words together and reach the following relation:

$$f9(s_i) = \frac{\sum w_{ki}}{|s_i|} \tag{42}$$

All the properties obtained from f1 to f9 are between 0 and 1. By combining them linearly and weighting the most important properties, the value of the function of belonging to a sentence to the summary can be determined to obtain. The initial population of sentences (countries) starts the imperialist competitive algorithm. This affiliation function is calculated as follows:

$$Summary_i = \left(\widetilde{s_{i1}}, \widetilde{s_{i2}}, \dots, \widetilde{s_{iN}}\right), \forall i = 1, N_{summary}$$
$$\widetilde{s_{ij}} = \left(\widetilde{s_{ij}}, \mu_{s_{ij}}\right), \forall j = 1, N \qquad (43)$$

N is the total number of sentences in the texts. The amount of the function of belonging of the sentence to the summary (initial country) is calculated as follows:

$$\mu_{s_{ij}} = \frac{a_1 \times f1(s_{ij}) + a_2 \times f2(s_{ij}) + a_3 \times f3(s_{ij}) + a_4 \times f4(s_{ij}) + a_5 \times f5(s_{ij}) + a_6 \times f6(s_{ij}) + a_7 \times f7(s_{ij}) + a_8 \times f8(s_{ij}) + a_9 \times f9(s_{ij})}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9} \qquad (44)$$
$$0 \le a_i \le 1$$

Now the algorithm starts running.To achieve coherent and readable summaries without duplicate sentences, objective functions were defined, and the imperialist competitive optimization algorithm was modified in multi-objective to achieve such summaries better, A special sensory-mental method was proposed to reach the target quickly and leave the optimal local range in the response space using density detection based on the normal probability distribution function of the response vectors.

Table 1
Comparison of *ICA* and *PSO* Algorithms in Summarizer System

	ROUGE-N	F1	Recall	Precision	Time Consumed
ICA	0.94	0.83	0.91	0.93	650
PSO	0.58	0.49	0.56	0.48	480

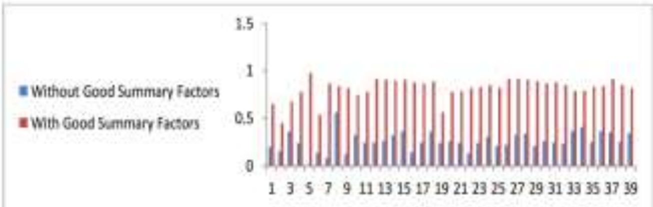

Fig. 1. Difference between Normalized Percentage of Optimality of Summaries Using Multi-objective Method and without it

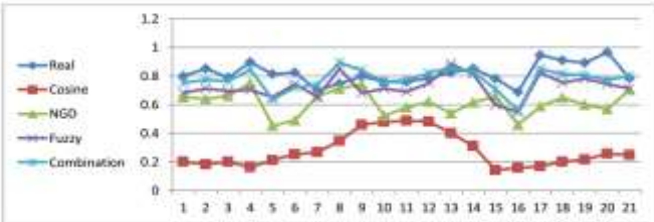

Fig. 2. Comparison of Sentences Similarity Methods

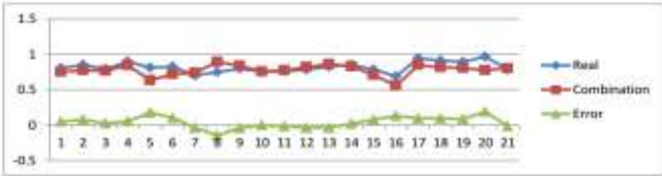

Fig. 3. Combined Similarity Criterion with an Average Error of 0.0195

7. Conclusion

In this research, different methods were used for this problem to optimize it; as used in the evaluation section, the database can be used according to the relationships of synonyms and contradictions between the words defined in it and the existing methods. We defined the factors that lead to optimizing the summary as goals that led to the production of summaries closer to the summaries produced by a human expert. Summaries were produced in the program with or without these objectives that showed differences. As can be seen, if the goals are used in the production of the summary, they are much more similar to summaries produced by the human expert. The absorption policy used, which differs from the usual method of imperialist competition, made the algorithm converge faster.

References

[1] B. Mohammadi, “Extraction of Key Points from English Texts using Frame Network Ontology,” in Proceedings of the 17th Annual National Conference of the Iranian Computer Association, Tehran; 2011.

[2] H. Bashiri, M. Shameh, “Using Clustering and Summarizing Documents to Distributed Hidden Semantic Indexing,” in Proceedings of the Third Data Mining Conference, Tehran; 2009.

[3] C.-Y. Lin, G. Cao, J. Gao, J.-Y. Nie, “An information-theoretic approach to the automatic evaluation of summaries,” in Proceedings of HLT-NAACL‘06, pp 463-470, Morristown, NJ, USA; 2006.

[4] Ch. Jung, R. Datta, A. Segev, “Multi-document summarization using evolutionary multi-objective optimization, GECCO,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion; 2017, pp. 31-32.

[5] N. Vanetika, M. Litvaka, E. Churkina, M. Lastb, “An Unsupervised Constrained Optimization Approach to Compressive Summarization,” Information Science; 2019, 509: 22-35.

[6] H. Mirshojaei, B. Masoomi, “Text Summarization Using Cuckoo Search Optimization Algorithm,” Journal of Computer & Robotics; 2015, 8(2): 19-24.

[7] A. Zamanifar, O. Kashefi, “AZOM: A Persian Structured Text Summarizer, NLDB 2011,” LNCS;2011, 6716, 234-37.

[8] A. Sh. Shahabi, M. R. Kangavari, “A Fuzzy Approach for Persian Text Segmentation Based on Semantic Similarity of Sentences,” Intelligent Information Processing III, Vol. 228 of the Series IFIP International Federation for Information Processing; 2006, pp. 411-420.

[9] D. Debnath, R. Das, P. Pakray, “Extractive single document summarization using multi-objective modified cat swarm optimization approach: ESDS-MCSO, Neural Computing, and Applications,” Neural Computing and Applications; 2021. https://doi.org/10.1007/s00521-021-06337-4.

[10] H. Mirshojaei, B. Masoomi, “Text Summarization Using Cuckoo Search Optimization Algorithm,” Journal of Computer & Robotics; 2015, 8(2): 19-24.

[11] J. Sanchez-Gomez, M. Vega-Rodriguez, Ch. Perez, “A decomposition-based multi-objective optimization approach for extractive multi-document text summarization,” Journal of Applied Soft Computing; 2020, 91: 106231.

[12] R. Alqaidi, W. Ghanem, A. Qaroush, "Extractive Multi-Document Arabic Text Summarization Using Evolutionary Multi-Objective Optimization With K-Medoid Clustering," IEEE; 2020, 8: 228206-24.

[13] A. Jangra, S. Saha, A. Jatowt, M. Hasanuzzaman, “Multi-Modal Supplementary-Complementary Summarization using Multi-Objective Optimization,” SIGIR '21: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information; 2021, pp. 818–828. https://doi.org/10.1145/3404835.3462877

[14] P. Wilson, J. Jeba, “A developed framework for multi-document summarization using softmax regression and spider monkey optimization methods,” Journal of Soft Computing; 2022. https://doi.org/10.1007/s00500-021-06694-1

[15] D. R. Radev, K. McKweon, “Generating natural language summaries from multiple on-line sources,” Computational Linguistics; 1998, 24(3): 469-500.

[16] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, et al., “MEAD – a platform for multidocument multilingual text summarization,” In Proceedings of LREC 2004, Lisbon, Portugal; 2004.

[17] M. Abel Fattah, F. Ren, “GA MR FFNN PNN and GMM based Models for Automatic Text Summarization,” Computer Speech, and Language; 2009, 23: 126-44.