Computers, Materials & Continua
DOI:10.32604/cmc.2021.013836
Article

Tech Science Press

Automatic Text Summarization Using Genetic Algorithm and Repetitive Patterns

Ebrahim Heidary[1], Hamïd Parvïn[2,3,4,*], Samad Nejatian[5,6], Karamollah Bagherifard[1,6], Vahideh Rezaie[6,7], Zulkefli Mansor[8] and Kim-Hung Pho[9]

[1]Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran
[2]Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam
[3]Faculty of Information Technology, Duy Tan University, Da Nang, 550000, Vietnam
[4]Department of Computer Science, Nourabad Mamasani Branch, Islamic Azad University, Mamasani, Iran
[5]Department of Electrical Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran
[6]Young Researchers and Elite Club, Yasooj Branch, Islamic Azad University, Yasooj, Iran
[7]Department of Mathematics, Yasooj Branch, Islamic Azad University, Yasooj, Iran
[8]Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[9]Fractional Calculus, Optimization and Algebra Research Group, Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[*]Corresponding Author: Hamid Parvin. Email: parvin@iust.ac.ir
Received: 30 August 2020; Accepted: 14 September 2020

Abstract: Taking into account the increasing volume of text documents, automatic summarization is one of the important tools for quick and optimal utilization of such sources. Automatic summarization is a text compression process for producing a shorter document in order to quickly access the important goals and main features of the input document. In this study, a novel method is introduced for selective text summarization using the genetic algorithm and generation of repetitive patterns. One of the important features of the proposed summarization is to identify and extract the relationship between the main features of the input text and the creation of repetitive patterns in order to produce and optimize the vector of the main document features in the production of the summary document compared to other previous methods. In this study, attempts were made to encompass all the main parameters of the summary text including unambiguous summary with the highest precision, continuity and consistency. To investigate the efficiency of the proposed algorithm, the results of the study were evaluated with respect to the precision and recall criteria. The results of the study evaluation showed the optimization the dimensions of the features and generation of a sequence of summary document sentences having the most consistency with the main goals and features of the input document.

Keywords: Natural language processing; extractive summarization; features optimization; repetitive patterns; genetic algorithm

1 Introduction

Given the increasing spread of electronic text documents, acquiring important features and information, and accurately utilizing such resources in the fastest and shortest possible time has been one of the main challenges for researchers in recovering digital text information. A fundamental solution to this challenge is the production and use of automatic summarization tools for generating a summary document having main features and information of the input document.

Text summarization is a process receiving the original text from the input source and producing its most important contents in a compact format based on user needs [1–3]. In another definition, summarization is the process of text revision and means to reduce the length of the original text using to show important expressions and sentences in such a way as to cover the main information and purposes of the original document [2]. For a summary produced by an automatic summarization, computer-based and machine tools are used to produce a summary document's content, in a way that it is always different from the human-generated summary because human have a deeper understanding of the syntactic and semantic structure of text document content compared to machine while the creation of this understanding is complicated and difficult for a machine [3].

The history of the automatic summarization activity dates back to 1950 [4,5]. One can refer to an important research in that decade in order to produce a summary of scientific documents using the frequency of features such as vocabulary and important phrases [6]. The automated summarization process has been one of the fields of interest for researchers in science and so far, numerous studies have been devoted to the production of tools and methods for summarizing texts in different languages. Nevertheless, it has always been a significant challenge.

In general, the production of a summary document is done in two ways: extractive summarization and abstract summarization. In extractive summarization, important sentences are selected from the original text without alteration and are exactly written in the summary text. The main advantage of this method is the simplicity of the selection system, whereas the primary disadvantage of this method is the lack of assurance in the coherence of the summary text. Abstracts summarization, concepts, and interpretations extracted from the original text are written in the summary document, and the sentences of the summary document are not necessarily the main sentences of the input document.

The techniques used in extractive summary include three statistical, linguistic and combination methods [3]. The statistical method uses the statistical distribution of specific features such as position, sentence length, or event related to the vocabulary in the document. In linguistic mode, there is a need for profound knowledge and linguistic knowledge, and the digitizer system is able to interpret, analyze, and then select important sentences of the input text. The hybrid method is a combination of the two previous methods used to generate summary. By studying the results of various methods, the use of a combination of extraction summary techniques is very effective in the quality of the produced summary. In this research, using the combined method and repetitive patterns, we optimize the vector of input text features to produce the summary document as best we can. Then, using the optimized features and applying the genetic algorithm, we strive to generate a summary document that includes all the important features and sentences of the original document. In this way, the final summary produces unambiguously, with the utmost precision, cohesion and coherence.

The structure of this research is as follows: the second section copes with a review of the literature of summarizing Persian texts and other languages. In Section 3, the proposed method

will be described. In the fourth section, the implementation and evaluation of the proposed method are addressed and in the fifth part the conclusion and suggestions are addressed.

2 Review of Related Literature

Research in the field of summarization began by Loon, 1958, which refers to the distribution of words in sentences and specification of the keywords of the document. Then, a variety of approaches were proposed in the field of summarization, based on different approaches. Most new methods are presented to improve previous methods. The proposed method by [4] is used to improve the automatic compilation of Persian texts, natural language processing and correlation graphs, the coefficient of influence of different grammatical roles using the TF-IDF method for the entire input text, and then the vector of sentences. The input text is based on the four features, the similarity of each sentence with the title of the text, the similarity of each sentence with the keywords, the degree of similarity of the two sentences together, and the position of each sentence. Then, using the similarity graph scrolling, sentences are selected based on the highest external likeness and the lowest internal resemblance for the summary text. By [7,8], Ferdowsi University of Mashhad, AJAZ system is presented for summarizing single syndicated Persian texts. The system consists of various factors such as the amount of similarity to the field, the effect of the words, the length of the sentences, the position of the sentence in the text, the important terms, the effect of the pronouns, the expressions and the terms defining the sentences, the specific expressions, and the degree of similarity with the title of the text to determine. The importance of the sentences of the input document is used. The linear combination of the above parameters is considered for determining the final value. In the selective summarization of Persian texts based on the PSO clustering algorithm [8], first all the sentences of the input document are extracted, and then the candidate terms of each sentence are specified. The Context-Vector for each candidate word is created and stored for one time. For each input text, first their sentences are determined, and then for each sentence, the similarity is calculated with respect to the above vector, and stored in the similarity matrix. Finally, using clustering algorithms, the sentences of each cluster are determined and then in each cluster, its important sentence is extracted according to the calculated scores.

The FarsiSum Persian Automatic Summarization System, presented in [9], is a modified version of the SweSum of Swedish texts [10,11]. The above system receives inputs in HTML format. The graph-based algorithm for summarizing Persian text [11], based on graph theory, selects the most important sentences of the input document. In this algorithm, nodes and edges are identified with different weights, and then the final weights of each sentence are determined by combining these values. The final weights indicate the significance of the sentence and the probability of displaying it in the final summary document.

Automatic summarization of online debatable texts using the vector space model [12–15] is suggested to produce a summary system for online debates using Abstract Summarizer. The extracted subject is based on an analysis of dependency and syntactic structure. The proposed system is implemented on the basis of three different modules: subject extraction, subject selection and summary generation. This system selects the theme tips and tips for the summary. After the selection process, shorter points are produced by smaller indirect points.

The proposed model is a text summarization extraction method [3], to find appropriate sentences using the statistical and linguistic approach. The summary system consists of three main stages of preprocessing, extracting sentences and a genetic algorithm for ranking sentences based on attribute weights. Each sentence is represented by a sentence property vector. For each

sentence, the linguistics statistical characteristics are examined. Each sentence has a score based on the weight of the features that are used to rank. The values of sentences are between zero and one. Using GA, the best chromosome is selected after a certain number of generations. Then, using the Euclidean distance formula, the distance between the score and the most appropriate chromosome is evaluated. Sentences are arranged in ascending order of distance. Depending on the compression, sentences are extracted from the document to produce a summary document. The HMM-based extraction summary system based on the Hermetic Markov Model (19) is a compression technique based on Part of speech (POS) tagging. Speech tagging is an automated tagging process for each word in a sentence. The automatic tagging of speech components is a machine learning technique that has been respected by researchers over the past two decades. Each natural language includes parts of speech such as verb, noun, adjective, and so on. In summarization model based on Genetic Algorithm, a hybrid learning approach based on genetic algorithm and a probabilistic method for selecting features is proposed. The proposed developed method considers five characteristics of the text, including similarity to the title, length of sentence, position of the sentence, numbers, figures and thematic words. Given the specific number of features used, the chromosomes consist of five genes and each gene is represented by a binary format feature.

The system provided by [13,14], known as the QueSTS system, is an integrated extracting summary query system from the set of documents. This approach represents an integrated graph of the relationships between the sentences of all the inputs. Existing relationships and subcategories are constructed from the original graph. Subcategories contain very relevant queries and are very closely related. These subcategories are ranked based on a scoring model. The highest ranked subcategory, which has a lot to do with query information, is selected as a special summary.

3 Proposed Method

In this research, a new hybrid method is proposed for extraction of single document texts based on the reduction and optimization of the dimensions of the input document features by identifying and extracting repetitive patterns. In the proposed method, using the vector of optimal features and using the genetic algorithm, a sequence of input document sentences is selected and a summary text is generated that covers all the main goals and characteristics of the input document. The proposed summarization process consists of three preprocessing steps, the optimization of the vector of attributes by creating repetitive patterns and summarizing, see presented Fig. 1. In the following, we will review the architecture of the proposed method.

The main advantage of the proposed idea is to optimize the feature vector, as well as the high precision in choosing the sentences of the original document for generating the summary document. In this method, we solve the problems of inconsistency and ambiguity to the desirable level in the summary document.

3.1 Preprocessing

Before performing the summarization operation, in order to produce a more precise summary document, the input text needs to be transformed into a single unit for the processing of the digest operation. Preprocessing operations include operations such as aligning, segmenting, tagging components of the word, rooting, and deleting the stop terms. For Persian language texts, one of the tools used at this stage is the ParsProcessor tool. ParsProcessor is a product manufactured and presented by the National Center for Cyberspace (Iran Telecommunication

Research Center). The evaluation of this tool shows a precision of 98% for tagging and a 100% precision for normalization.

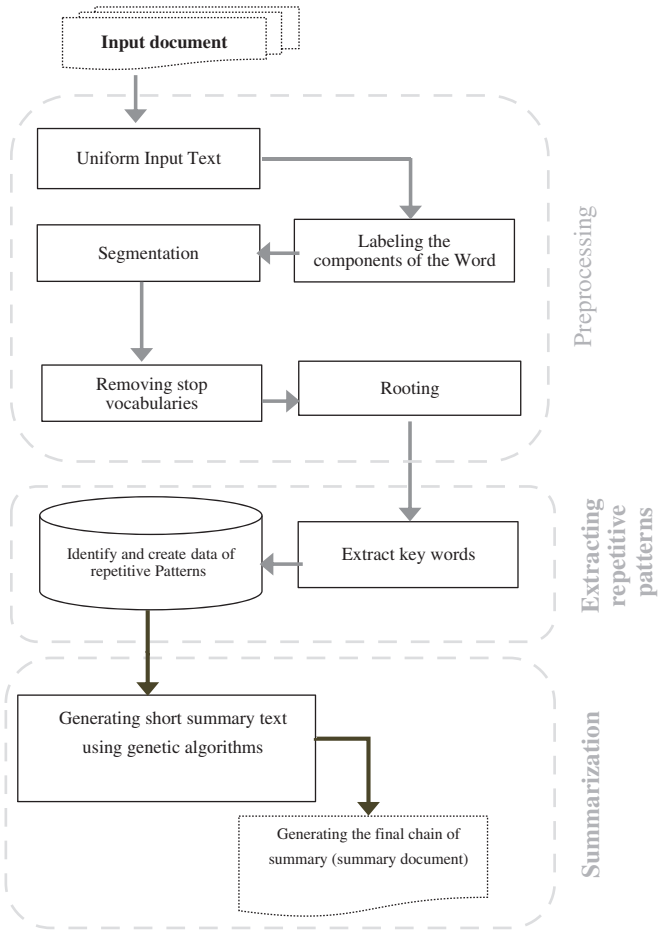

Figure 1: The new proposed summarization method

3.1.1 Homogenizing Input Text

The preprocessing process involves standardizing and homogenizing the original text. One of the main problems in the interpretation of the texts, including Persian texts, is a number of forms in the same vocabulary. This has caused problems in identifying the same vocabulary. To solve this problem, in the first step, we will align the input text corpus.

3.1.2 Tagging the Components of the Word

After performing the normalization operations in the previous step, at this stage, the role of the term (such as: Noun, verb, adjective, conjunction, etc.) in the sentence is tagged for use in other tagging steps.

3.1.3 Segmentation

In the segmentation step, the input document is identified using the marks separating the boundaries of terms and sentences. It should be noted that segmentation operations are performed to identify words and sentences of the original text. It is worth noting that the boundary of sentences is determined by verbs or conjunctions, or by other attributes.

3.1.4 Removing Stop Words

Stop words are vocabularies that are commonly used in text documents, but not descriptive, and not related to a specific subject. These words do not have meanings and include conjunctions and linking verbs, pronouns, prepositions, and types of adverbs.

3.1.5 Rooting

In the rooting stage, we identify and extract the roots of the words and lexicons in each sentence from the input text.

3.2 Summary Operations

In the proposed method, the main operation of selecting sentences and producing the text of the summary document is done in this phase. In this method, by identifying the relationship between the key words of different texts from different classes of subjects (sports, economic, political, scientific, etc.), we create the data of repetitive patterns and then use the genetic algorithm and exploit the repeated patterns of extraction in the input document and generate the summary text with the greatest similarity to the original text, see Fig. 2 below.

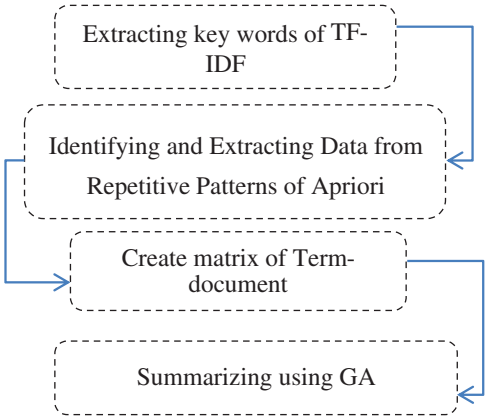

Figure 2: Summarization operations

3.2.1 Extracting Key Words Using TF-IDF

The TF-IDF method is widely used for weighting. Initially, this method is introduced to retrieve information. TF-IDF (Term Frequency-Inverse Document Frequency), depending on the frequency of the document itself, weighs a term [14–24]. This means that if the documents display more than one term, then that term will be less important and its weight will be less. The method is described in Eq. (1):

$$a_{ij} = tf_{ij} * \log\left(\frac{N}{n_j}\right) \tag{1}$$

In the above formula, tf_{ij} is the frequency of the j statement in i, n denotes the total number of documents in the data, and n_j represents the number of documents that has term i. When N equals n_j, then a_{ij} is zero, which is often displayed in small datasets, so we need to apply some uniformity techniques to improve the formulas, see Eq. (2):

$$a_{ij} = \log\left(tf_{ij}\right) * \log\left(\frac{N+1.0}{n_j}\right) \tag{2}$$

The ease of use of this method is why the method is more acceptable than other methods. Therefore, with regard to the existence of classified political, sporting, and economic texts, etc., keywords are obtained using the Eq. (1) in each class of texts, with the elimination of repetitive words in each class, and stored in a data set.

3.2.2 Creating Dataset of Repetitive Patterns Using the Apriori Algorithm

Apriori is an ordinary algorithm introduced for the first load in order to extract associative rules. There are two steps to extract association rules: (1) Recognition of repetitive items; (2) Creating community rules of repetitive items. Repetitive items can be extracted in two steps. First, the candidate items are generated, and then sets of items are extracted with the help of candidate items. Repetitive item support is nothing more than the minimum user-specified support [16–19]. The Apriori algorithm can be seen as presented in Fig. 3.

```
Ck : Candidate itemset of size k
Ik : Frequent itemset of size k.
I1 ={l arg e 1 –itemsets};
for (k = 2; I k-1 ≠ 0; k ++) do begin
Ck=apriori –gen(Ik-1);
 //New candidates for all transactions
T ∈D do begin
CT=subset(Ck ,T);
//candidates contained in T
For all candidates c ∈CT do
c.count ++;
end;
end;
 Ik={c ∈ Ck | c.count>=min sup}
end
Answer= Uk Ik;
```

Figure 3: Apriori algorithm

3.2.3 Extracting Repetitive Patterns

Now, to use the Apriori algorithm and to extract repetitive patterns in the texts, the keywords obtained from the previous step are considered as the item and each text is also a transaction; then, in each text, it is examined which of the words is a keyword. After identifying the keywords, we put one under the columns containing the keyword and zero under the rest, as in the table in Fig. 4.

Then, in order to find repetitive patterns that have been frequently repeated in all texts, they must be converted to the apriori input format according to Fig. 5, and set up the itemset.

The next step is to enter the mafia algorithm that recognizes repetitive patterns, and we find repetitive patterns that are repeated in all texts very much.

Subject	Doc #	KEYWORD					
		Mesi	Football	Barselona	Javi	News	...
Sport	D1						...
	D2						...
	...	...	...	...	...	...	...

Figure 4: Keyword table

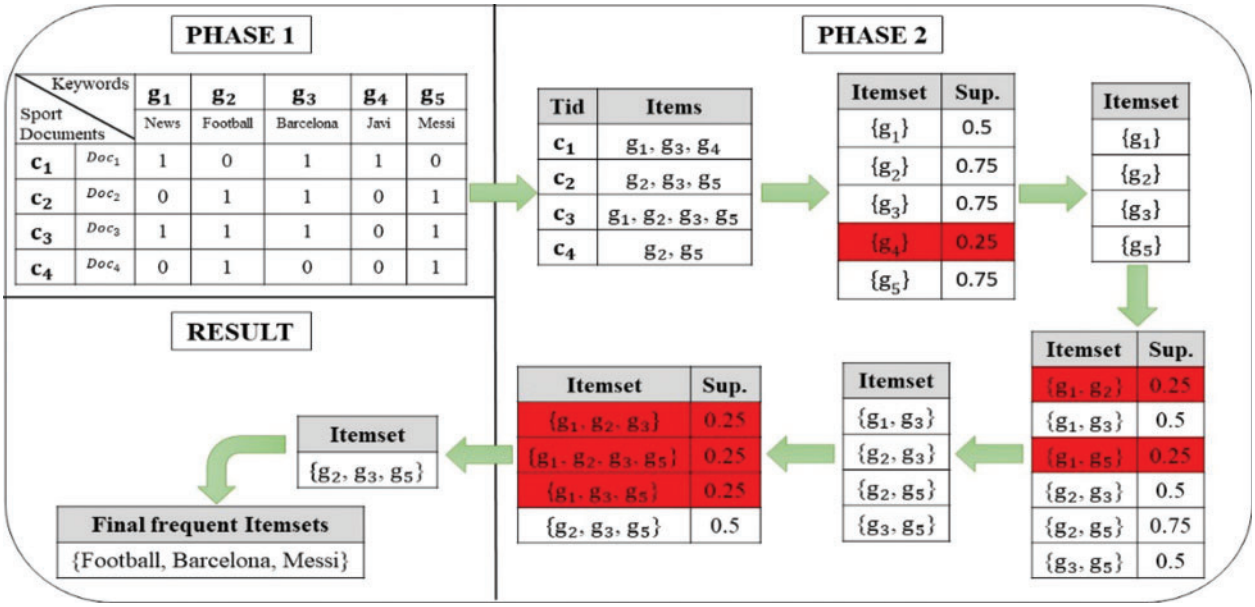

Figure 5: Extraction of repetitive patterns

As we see in the above, in Phase 1, the key word table is made according to the format introduced for input to the Apriori algorithm. In Phase 2, we create a collection of repetitive patterns. In this template, we set the threshold value to 0.5. If the repeat count of a set of repetitive patterns is lower than the threshold value, then this process continues until the final repetitive pattern is detected. Here, the number of repetitive patterns of sports texts may be n. If the keywords of all sports texts k have been repeated, the repetitive patterns between the k keyword n have been identified by the repetitive pattern ($n > k$). For example, in sports texts, $n = 700$ and $k = 20$, this has greatly reduced the number of features. For example, Esteghlal, Schaeffer, League has been identified as a repetitive pattern for sporting documents. After we have obtained repetitive patterns, we introduce a repetitive pattern in the new set. We repeat the

same process for documents of other topics. Now assume that for political documents or texts, for example, the number p has been found to be repeated, we add that number to the repetitive pattern set, if it is a repetitive pattern, we delete a set that is unlikely to be common in political and sports patterns. A repeat pattern is found (the following Eq. (3)).

$$SUM_{keywords} = (|p| + |k|) - |p \cap k| \tag{3}$$

The point is that if there were repetitive patterns between the two different problems, we would generally eliminate the set because if the pattern is both political and sports, it does not matter and does not have information load.

In the end, we consider the selected features as repetitive patterns that we have in one set. For example, there are 20 ones for sports, 25 ones for politics, and 14 ones for economics, i.e., a total of 100 ones as a repetitive pattern.

Now, every text in any domain is considered as a record and columns are considered as repetitive patterns.

3.2.4 Term-Document Matrix

The Term-document matrix is an incidence matrix, which describes the number of occurrences occurring in the terms, in a set of documents. In this matrix, the rows of the matrix represent the documents and columns represent the terms. There are various ways to determine the value of each entry in a matrix, in which we use a binary vector. In the column section of the matrix, instead of placing all the terms, we use the repetitive patterns obtained from the preceding steps, so that if the document i contains a repetitive pattern j, then $w_{ij} = 1$, otherwise $w_{ij} = 0$ and as in Fig. 6, a complete dataset of repetitive patterns is generated for all test documents in different subject categories.

Subject		FIS_1	FIS_2	FIS_3	FIS_4	…..
Sport	D_1	0	0	0	1	
	D_2	1	0	0	0	
	…..					
	D_m	1	1	0	1	
Economical	D_{m+1}	1	0	0	1	
	D_{m+2}	0	0	1	1	
	…..					
	D_n	0	1	1	0	
….	D_{n+1}	1	0	0	1	
	…..					

Figure 6: Term-document matrix generation

3.3 Summarization Using a Genetic Algorithm

Genetic algorithm is a special type of evolutionary algorithm that uses evolutionary biology techniques such as heritability, biology mutation, and Darwinian selective principles to find the optimal formula for predicting or matching the pattern. Genetic algorithms are often a good

option for regression-based prediction techniques. In modeling the genetic algorithm is a programming technique that uses genetic evolution as a problem solving model. The problem that needs to be solved is the inputs that are converted into solutions through a process of genetic evolution. Then, the solutions are evaluated as candidates by the Fitness Function, and if the extraction condition for the problem is provided, the algorithm ends. We use the features of this algorithm in this paper and by generating different generations of a document, we will get the best summary text of the temporary document based on a cost function.

3.3.1 Temporary Summary Document Generation

The nature of the genetic algorithm is a repetition-based algorithm, most of which are selected as random processes, which are composed of parts of the fitting, display, selection and modification functions. Before a genetic algorithm runs for a problem, a method for coding the genomes into a computer language should be used. One of the usual ways to code is binary strings: strings 0 and 1. We used this method in this article.

In each problem, before the genetic algorithm can be used to find an answer, two elements are needed: First, a method is needed to provide an answer in the form in which the genetic algorithm can function on it. Second, in the fundamental component of the genetic algorithm, there is a method that can calculate the quality of each proposed solution using fitness functions, which is discussed below.

In this paper, we used genetic algorithms to randomly generate several temporary summary documents for the problem and call it the primitive population and consider each document as a chromosome, the length of the chromosomes here being equal to the number of sentences of the text of the original document. During each generation, each feature of fitness value is evaluated by the fitting metric function chosen in this study by the cosine function. After selecting the best chromosomes, we combine the chromosomes together and make a mutation in them. Finally, we combine the current population with a new population that results from the combination and mutation in the chromosomes, as in Fig. 7. In this algorithm, it is clear that in each generation, the most suitable ones are selected not the best ones.

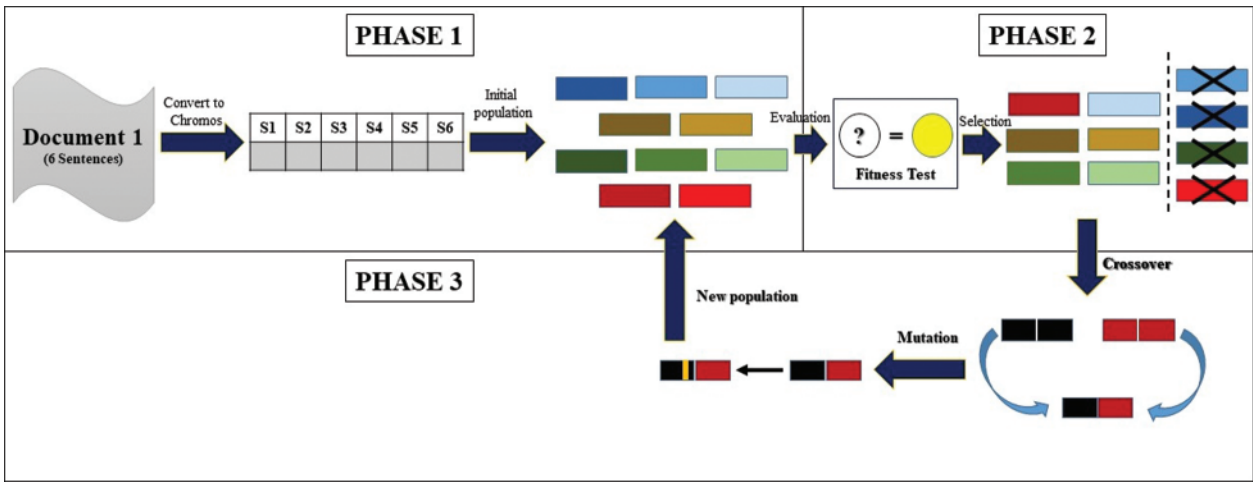

Figure 7: Summarization process architecture using genetic algorithm

We used the above process to create a new generation of chromosomes and different from the previous generation in order to be able to extract the most appropriate summary document. The entire process is also repeated for the next generation, and the pair of documents are selected for composition, the third generation population is created, and this process is repeated until one of the conditions for termination of genetic algorithms, such as reaching a constant number of generations or the completion of time of the dedicated calculation will occur.

For example, suppose the text of sport 1 contains 10 sentences whose Term-document matrix had been obtained in terms of repetitive patterns in the previous section; now, using the above algorithm, we extract the best summary text for a single text in a genetic method.

Phase 1: Now, in accordance with phase one of the above algorithm, a chromosome with 8 genes should be produced which is then generated using the binary crash function and then the initial population of the algorithm for text 1 is generated. In fact, each of the chromosomes represents a summary document that randomly selected sentences of the text and compiled the summary text. For all chromosomes produced, a Temp Vector was created based on repetitive patterns (Fig. 8).

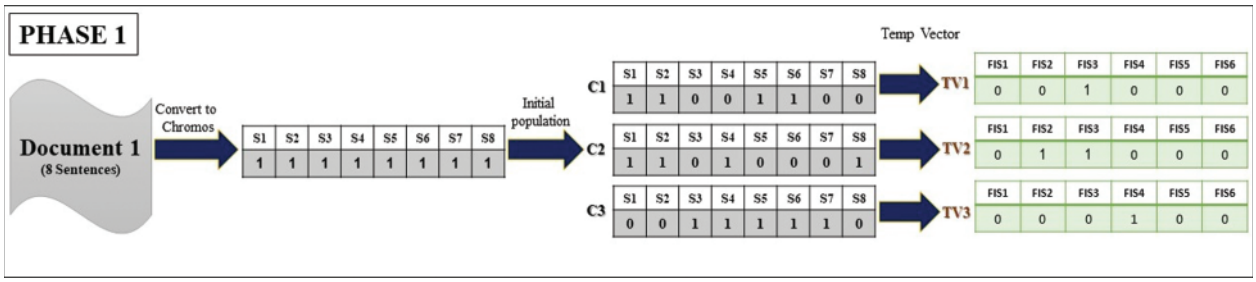

Figure 8: The development of temp vectors in terms of repetitive patterns

Phase 2: The second phase of the above algorithm is the most important phase which calculates the evaluation. In this phase, we have used the similarity of the cosine distance Eq. (4) to find the similarity between the original document and the summary documents. Based on the similarity of the two vectors, the relation between the two n-dimensional vector $d_j = (x_1, x_2, x_3, \ldots, x_n)$, for each vector, x_i represents the i-th sentence of the d_j document.

$$\cos(\theta) = \frac{d_1 \cdot d_2}{|d_1||d_2|} \tag{4}$$

The angle between the two vectors according to Eq. (4) will be between zero and one, and if the two vectors (text) are the same, the cosine distance obtained is one, and if the two vectors are completely different, the cosine distance value will be zero.

In this paper, we first obtained a Main Vector based on the Term Document for each original document D_i in terms of repetitive patterns and called it MV_i, Using the genetic algorithm, we generated generations from the original D_i document as temporary summary documents, and then generate a Temp Vector based on $TV_J = \{TV_1, TV_2, TV_3, \ldots\}$ repetitive patterns for each generation.

Using Eq. (4), we now calculate the similarity of MV_i vector to all TV_js generated from the temporary summary D_i documents as shown in Fig. 9. If a generation does not meet

the threshold of similarity, it will be excluded from the population and will not be used in future operations.

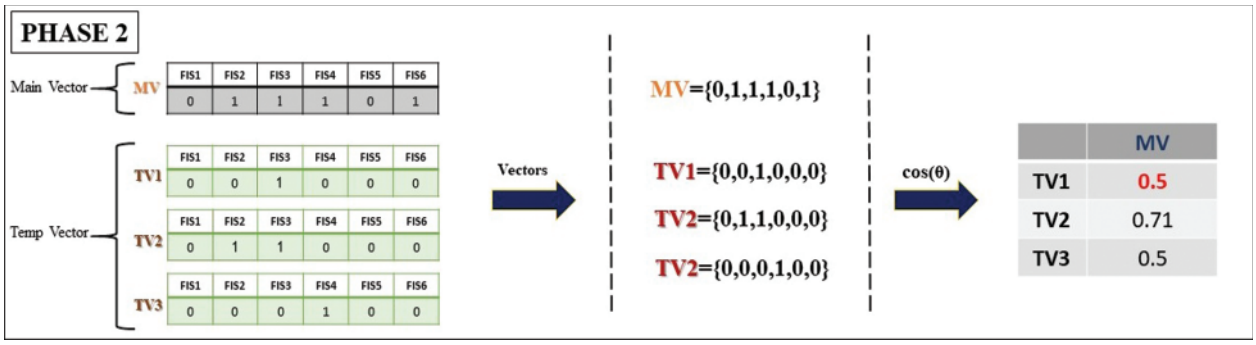

Figure 9: Computation and evaluation of temp vectors and the original document in terms of cosine similarity

After obtaining the angle of the vectors by the cosine similarity criterion according to Eq. (5), the best candidate is the level of similarity of the D_1 text with the chromosomes produced for it, C_1 and C_3, because they have the smallest distance. So, it is possible that up to this stage, C_1 will be the candidate for the best summary text for D_1 document.

$$\cos(\theta) = \frac{d_1 \cdot d_2}{|d_1|\,|d_2|} = \frac{1\times1+1\times1+0\times0+0\times1+0\times0+0\times0+0\times0+0\times1}{\sqrt{1^2+1^2}\times\sqrt{1^2+1^2+1^2+1^2}} = \frac{2}{\sqrt{8}} = 0.71 \tag{5}$$

Phase 3: In order to be able to summarize the texts to maximize the semantic and structural similarities to the original document by creating new generations, we need to make changes in the selection of sentences and, as a result, new generations of that original text, so that we can resemble the similarity more precise in the summarized texts. For example, in the practice of combining, we have created two generations with the highest rank of similarity to the input document, randomly adding their sentences together, the composition, and the two new generations that we have reached for generations which provide a higher and more accurate comparative rating than their fathers. You can also create a mutation in a generation. For example, you can move two replace together and create a new generation.

3.3.2 Generating the Final Summary Document

In this research, the final summary document contains a chain of sentences that should be continuously generated without any ambiguity. It is noteworthy that in some sentences of the text of the summary document, there may be pronouns whose reference is unclear and may be ambiguous for its users.

By examining this problem, it has been found that in most cases, the reference of the ambiguous pronouns is in the preceding sentence. Therefore, to eliminate the ambiguity of an unspecified pronoun, the preceding sentence is rewritten from the original in the summary text.

4 Implementing and Evaluating the Results

In this research, the proposed method on the body of texts was evaluated from five different categories of topics (Tab. 1) and Java programming language for design and implementation.

In the proposed method, we first extracted the key words of each subject and then, using the relationships between key words and the Apriori algorithm, repetitive patterns were identified among key words in different topics, and a set called dataset of repetitive patterns were formed.

Table 1: The number of subject templates

Subject	Number of documents
Sport	1000
Economics	1000
Politics	1000
Science	1000
Others	1000
Total number of documents	5000

One of the main advantages of this method is to optimize the feature vector using dataset of repetitive patterns. After generating the data, repetitive patterns are generated using the genetic algorithm of the summary document.

In this method, using the genetic algorithm, we first randomly generate several temporary summary documents for the problem, which are called initial populations, and we consider each document as a chromosome, whose chromosome length here is equal to the number of sentences in the document text. During each generation, each feature with fitness value is measured and evaluated using the cousinial fitness function.

After choosing better chromosomes, we combine the chromosomes together and make a mutation in them. Eventually, the current population will also be generated by a new population of combinations and mutations in the chromosomes. In this method, in each generation, the most appropriate document is selected, which is less different from the original document.

To evaluate the proposed method, using the Precision and Recall criteria, we evaluated the results and efficiency of the proposed summarizer method. These criteria are defined on the basis of the following Eqs. (6) and (7):

$$\text{Precision (P)} = \frac{|\text{Sum}_r \cap \text{Sum}_s|}{|\text{Sum}_s|} \tag{6}$$

$$\text{Recall (R)} = \frac{|\text{Sum}_r \cap \text{Sum}_s|}{|\text{Sum}_r|} \tag{7}$$

where Sum_r is a human-produced summary and sums is the summary document generated by the summarizer. Combining the results of the criteria for evaluating the precision and recall of the proposed method is calculated using the following Eq. (8):

$$E_{p,r} = \frac{2pr}{p+r} \tag{8}$$

The results of the evaluation of the proposed method are presented in the following Tab. 2, based on the evaluation criteria.

Table 2: The comparison between the proposed method with the existing ones

Method	P	R	$E_{p,r}$
SweSum	0.66	0.69	0.67
proposed method	0.73	0.75	0.74

Given the results of the above table, the proposed summarization method has a higher precision than the SweSum summarizer with a similar set of data, see Fig. 10 below.

In order to evaluate the effectiveness of the proposed method in generating a desirable summary document, readers' satisfaction with the summaries produced by the proposed method with various topics and was examined and that the documentation of the summaries generated by 5 experts were reviewed as well. Below is the average score of satisfaction feedback from readers (Fig. 11).

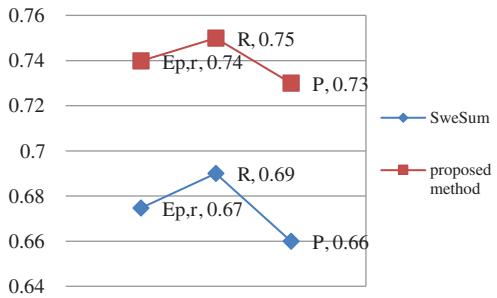

Figure 10: The results and efficiency of the proposed summarizer using precision and recall criteria

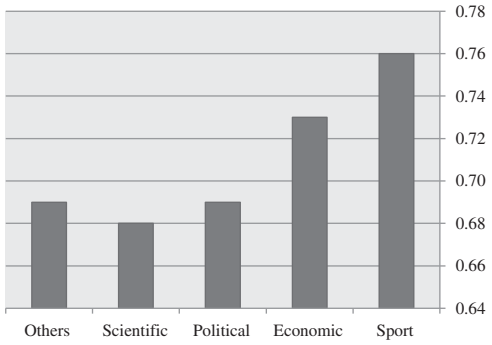

Figure 11: Assessment of readers' satisfaction from summaries of various topics

The other evaluation criterion used is the average of the common sentences of human summaries compared to the proposed method. For further explanation of the evaluation of the proposed method, the best and the weakest human-made document (by 5 experts) and the proposed synthesizer are seen below (Tab. 3).

Table 3: A comparison between the proposed method with the human summarizer. (A) The best document produced by the proposed method (The original document has 35 sentences). (B) The weakest document generated by the proposed method (The original document has 28 sentences)

(A)		
Presence or lack of presence in the proposed method	Percentage of presence in the summary by human	The number of selected sentence
√	100	1
√	100	3
√	90	4
√	50	6
√	80	7
×	70	9
×	70	11
√	50	12
×	40	15
√	40	18
×	40	21
×	30	25
√	50	26
√	80	29
√	80	32
(B)		
√	100	1
×	100	2
√	100	3
×	80	5
√	40	8
×	60	11
√	50	12
√	50	14
×	80	17
√	80	20
×	60	21
×	90	23
√	80	25

As the results show, the proposed method has chosen the most precision for selecting the important sentences of the original document in the production of a summary document selected by expert individuals.

5 Conclusion

In this paper, we proposed a novel method for selective text summarization using genetic algorithms and generating repetitive patterns. One of the important features of the proposed method in comparison with other previous methods is to optimize the vector of the main document's properties in the production of a summary document by identifying and extracting the relationship between the main features of the input text and the creation of repetitive patterns.

In the proposed method, in the process of producing a summary document, using genetic algorithms, randomly, several temporary summary documents are generated for the problem, and are called initial populations, and we consider each document as a chromosome, in which is the length of the chromosomes here is the number of sentences in the text of the original document.

During each generation, any feature with fitness value is evaluated using the cousinous measurement fit function. After choosing the best chromosomes, we combine the chromosomes together and make a mutation in them. Eventually, the current population will also be generated by a new population of the combinations and mutations in the chromosomes. In this algorithm, it is established that in each generation, the most appropriate summary document with less difference to the original document is selected.

In evaluating the proposed method, on the structure of 5 categories of the subject, using the Precision and Recall criteria, we showed that the proposed method has a higher precision of approximately 0.74 compared with the SweSum summarizer with similar data. Furthermore, the average satisfaction of readers from the summary documents generated by the proposed method in various subjects is estimated at 0.71 and it also shows the highest precision in choosing the important sentences of the original document that has been selected by experts in the production of the summary. The idea for the proposed method has overcome some problems such as inconsistency and ambiguity in the summary text. A combination of semantic communication techniques in the original document to improve the proposed summarization process is proposed for future research.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Pourmasoomi, M. Kahani, S. A. Toosi and A. Estiri, "Ijaz: An operational system for single-document summarization of Persian news texts," *Journal Signal and Data Processing*, vol. 11, no. 1, pp. 33–48, 2014.

[2] C. Salmet, A. Atmadja, D. Maylawati, R. Lestari, W. Darmalaksana *et al.,* "Automated text summarization for indonesian article using vector space model," in *The 2nd Annual Applied Science and Engineering Conf.*, Bandung, Indonesia, vol. 288, no. 1, pp. 1–6, 2017.

[3] N. Mazdak and M. Hassel, "FarsiSum-A Persian text summarization." Master thesis, Stockholm University, Department of Linguistics, Sweden, 2004.

[4] M. G. Ozsoy, F. N. Alpaslan and L. Cicekli, "Text summarization using latent semantic," *Journal of Information Science*, vol. 37, no. 4, pp. 405–417, 2011.

[5] C. Thaokar and L. Malik, "Test model for summarization Hindi text using extraction method," in *IEEE Conf.*, vol. 7, no. 3, pp. 1138–1143, 2013.

[6] M. Bazghandi, G. H. Tedin-Tabrizi, M. Wafai-Jahan and A. Bazghandi, "Selective synopsis of PSO clustering coherent texts, " in *The First Int. Conf. on Line Processing and Farsi Language*, Iran: Semnan University, 2012.

[7] C. R. Chowdary, M. Sravanthi and P. S. Kumar, "A system for query specific coherent text multi-document summarization," *International Journal on Artificial Intelligence Tools*, vol. 19, no. 5, pp. 597–626, 2010.

[8] S. Lagirini, M. Redjimi and N. Azizi, "Automatic Arabic text summarization approaches," *Internatinal Journal of Computer Applications*, vol. 164, no. 5, pp. 31–37, 2017.

[9] A. D. Chowanda, A. R. Sanyoto, D. Suhartono and C. J. Setiali, "Automatic debate text summarization in online debate forum," *Elsevier Science Direct*, vol. 116, pp. 11–19, 2017.

[10] Z. Sarabi, H. Mahyar and M. Farhoodi, "ParsiPardaz: Persian language processing toolkit," in *3rd Int. Conf. on Computer and Knowledge Engineering*, Iran: Ferdowsi University of Mashhad, pp. 1–8, 2013.

[11] J. Steinberger and K. Jezek, "Evalauation measures for text summarization," *Computing and Informatics*, vol. 28, no. 2, pp. 1000–1025, 2014.

[12] G. Salton, A. Wong and C. S. Yang, "A vector space model for automated indexing," *Communications of the ACM*, vol. 60, pp. 1–8, 1975.

[13] L. Talibali and N. Riahi, "An overview of automatic text summarization techniques," in *Int. Conf. on Applied Research in Information Technology, Computer and Telecommunications*, New York, vol. 28, pp. 75–84, 2015.

[14] N. Riahi, F. Ghazali and M. Ghazali, "Improving the efficiency of the Persian abstract synthesis system using pruning algorithms in neural networks," in *The First Int. Conf. on Line and Language Processing Persian*. Iran: Semnan University, 2012.

[15] F. Mohammadian, M. Nematbakhsh and A. Naghshenilchi, "Summary of Persian texts using a meaning-based method," in *Second National Conf. on Soft Computing and Information Technology*, New York, vol. 7, no. 7, 2011.

[16] H. Sotoudeh, M. Akbarzadeh-Totouchi and M. Teshnelab, "Summary of text based on selection using an anthropological approach," in *18th Iranian Conf. on Electrical Engineering*, Iran: Isfahan University of Technology, vol. 1, pp. 2266–2227, 2010.

[17] M. Shamsfard and Z. Karimi, "The automatic writer system of Persian texts," in *12th Iranian Computer Society Conf.*, Iran, Tehran, vol. 40, pp. 1–28, 2006.

[18] H. Dalianis, "SweSum- A text summarizer for Swedish, Technical report in interaction and Presentation Laboratory," *Sweden*, pp. 1–15, 2000.

[19] H. Shakeri, S. Gholamrezazadeh, M. Amini-Salehi and F. Ghadmyari, "A new graph-based algorithm for Persian text summarization," *Springer Science and Business Media*, vol. 1, pp. 21–30, 2012.

[20] S. B. Rodzman, S. Hasbullah, N. K. Ismail, N. A. Rahman, Z. M. Nor *et al.,* "Fabricated and Shia Malay translated hadith as negative fuzzy logic ranking indicator on Malay information retrieval," *ASM Science Journal*, vol. 13, no. 3, pp. 100–108, 2020.

[21] M. M. Abdulnabi, R. Hassan, R. Hassan, N. E. Othman and A. Yaacob, "A fuzzy-based buffer split algorithm for buffer attack detection in Internet of Things," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 17, pp. 5625–5634, 2018.

[22] M. A. A. M. Zainuri, E. A. Azari, A. A. Ibrahim, A. Ayob, Y. Yusof *et al.,* "Analysis of adaptive perturb and observe-fuzzy logic control maximum power point tracking for photovoltaic boost DC-DC converter," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1, pp. 201–210, 2019.

[23] Z. M. Rodzi and A. G. Ahmad, "Fuzzy parameterized dual hesitant fuzzy soft sets and its application in TOPSIS," *Mathematics and Statistics*, vol. 8, no. 1, pp. 32–41, 2020.

[24] A. M. S. Bahrin and J. M. Ali, "Hybrid fuzzy-disturbance observer for estimating disturbance in styrene polymerization process," *Materials Science and Engineering Conference Series*, vol. 778, no. 1, 012089, 2020.