

---

# Automatic Persian text summarizer using simulated annealing and genetic algorithm

**Elham Mahdipour, Masoumeh Bagheri**

Computer Engineering Department, Khavaran Institute of Higher Education, Mashhad, Iran

**Email address:**

Mahdipour@khi.ac.ir (E. Mahdipour), M\_bagheri3000@yahoo.com (M. Bagheri)

**To cite this article:**

Elham Mahdipour, Masoumeh Bagheri. Automatic Persian Text Summarizer Using Simulated Annealing and Genetic Algorithm. *International Journal of Intelligent Information Systems*. Special Issue: Research and Practices in Information Systems and Technologies in Developing Countries. Vol. 3, No. 6-1, 2014, pp. 84-90. doi: 10.11648/j.ijjis.s.2014030601.26

---

**Abstract:** Automatic text summarization is a process to reduce the volume of text documents using computer programs to create a text summary with keeping the key terms of the documents. Due to cumulative growth of information and data, automatic text summarization technique needs to be applied in various domains. The approach helps in decreasing the quantity of the document without changing the context of information. In this paper, the proposed Persian text summarizer system employs combination of graph-based and the TF-IDF methods after word stemming in order to weight the sentences. SA-GA based sentence selection is used to make a summary, and once the summary is created. The SA-GA is a hybrid algorithm that combines Genetic Algorithm (GA) and Simulated Annealing (SA). The fitness function is based on three following factors: Readability Factor, Cohesion Factor, and Topic-Relation Factor. Evaluation results demonstrated the efficiency of the proposed system.

**Keywords:** Automatic Text Summarization, Stemming, TF-IDF, Genetic Algorithm, Simulated Annealing

---

## 1. Introduction

Nowadays with increase of information, users need to have access to effective methods in order to search for the requested information. In most cases, people study the summary of a document rather than the whole. Automatic text summarization is a solution for this issue. Automatic text summarization is a process to reduce the volume of text documents; using computer programs to create a text summary with keeping the key points and important documents. This approach shortens the information content of a text file while preserving the original contents [1]. Summarizing large documents is a difficult task for human. One approach to respond to the cumulate growth of information is to use automatic text summarization to decrease the volume of information and increase speed of accessibility to important notes.

Research studies on automatic text summarization started at mid-1950s [2] and is an old challenge that needs attention of text mining researchers in the field of computational intelligence, machine learning and natural language processing, and various methods such as neural networks, decision trees, semantic graphs, regression models, fuzzy logic, swarm Intelligence [3]. Research studies show that the

content of summaries depends on the input source, nature of the text, purpose and target of readers. Overall automatic text summarization consists of three main steps: 1) identifying keywords of the document as well as the most important pieces of information in the text; 2) interpreting the content of the document, extracting two or more topics, then combining them to one or more concepts; 3) creating a good summary with formulating the extracted sentences and combining them with the concepts [2].

In this paper we have proposed Phoenix Summarizer for development of Parsina summarizer [2, 4] and designed for improving the quality of text extraction. This paper is structured as follows. Section 2 introduces concepts of summarization. In section 3, the previous automatic text summarizer is reviewed. In section 4, the proposed automatic text summarization system, known as Phoenix, is introduced. Section 5 evaluates the proposed Phoenix system. Finally in section 6, the results and comparison of Phoenix and Parsina text summarization systems are discussed.

## 2. The Concepts of Automatic Text Summarization

The important factors in text summarization include: input factors, output factors, and target factors [5, 6].

Input factors are listed as follow:

- 1 Input form, such as:
  - a. Length of input text: short text (1 or 2 pages), long text (more than 50 pages).
  - b. Text structures: text structures that impact on text process such as: paragraph, entity, and predicate.
  - c. Text language: such as single language and multiple languages.
  - d. Text type: such as single document or multiple documents.
- 2 Text topic, such as:
  - a. Normal text: the subjects have vast knowledge domain such as scientific, sport, and farming.
  - b. Special text: the topics depend on knowledge of reader such as computer and medical.
  - c. Limited text: the special topics pertain to organization or society such as news and technical reports.

Output factors are listed as follow:

- 1 Content: this factor pertains to normal summary or response, extracted summary or abstract. Also when summary is normal, content included the most important information.
- 2 Format: text format may be having figures, tables, diagrams and etc.
- 3 Style: summary style can be predication or advisement.

The targets factors depend to field of application text and cause of summary and determined based on three factors:

- 1 Listeners: The level of listener's knowledge about the summary text field directly affects the results.
- 2 Conditions: the conditions or purpose determine means of the summary. When summary field is vast and identified, can remove the details.
- 3 Application: Depending on the application of the summary, the approach for creating summary text could change. As an example, will the summary be used for retrieving the whole text? Or is it a replacement for the input text? Or it could even be an overview of a reviewed document. In general terms, the application could be divided into public and interrogative categories.

In Addition, the summarizer system can be divided into three categories: [7]:

- 1 Category based on form and organization: Overall summaries based on form and organization can be divided into extraction and abstract. In extraction, summary based on statistical information of text, determination of the important sentences, and selection of the most important items to create the summary. The abstract methods attend to relationship between difference parts of text, terms concepts and semantics. For large documents, using extraction methods are easier and more flexible. However, the results might be

incoherent and un-related. On the other hand, the abstract method presents the summary of important parts and key points. In this method the compress rate is high due to summary hint to the content of text. Also, the resulting text would be coherent. Therefore abstract systems are stronger than extraction systems.

- 2 Category based on process level: we can divide summaries based on process level to superficial and deep categories. The superficial method, displays information based on superficial properties such as statistical conditions, place conditions, terms, and special domain. The summaries of this category are created by extraction method. The deep methods, employs natural language processing and needs to perform semantic analysis such as syntax and semantic relationships. The summary of this category creates combination of extraction and abstract methods.
- 3 Category based on addressee: the summaries based on addressee could be divided into three categories: public, based on query, and special topic. Purpose of public summary is for widespread society of readers and generally all topics are important. The application of summary based on query is related to one question such as "what is the cause of incidence?". The purpose in special topic summary is the user revenue and emphasis on special topic or single document and multiple document categories, or single language and multiple language categories.

### 2.1. Summarization Methods

There are many methods for text summarization such as information retrieval (IR) [6], clustering [8], graph theory [9], machine learning [10], Latent semantic analysis [6], and TF-IDF [2, 11]. We describe TF-IDF methods because the proposed system uses this because it is easy and accurate. Also the combination of TF-IDF and graph based methods is more applicable.

The TF-IDF weighting system is used with some adoptions which is similar to IR model. TF-ISF weights are computed for each sentence (equation (1) and (2)), where  $tf_{i,j}$  is term frequency of  $i^{th}$  index term in the  $j^{th}$  sentence, and  $isf_i$  is inverse sentence frequency of  $i^{th}$  index term. Also  $N$  is the number of all sentences and  $n_i$  is the number of sentences which contain  $i^{th}$  index term [2, 11].

$$tf_{i,j} = \frac{freq_{i,j}}{\max_i freq_{i,j}} \quad (1)$$

$$isf_i = \log \frac{N}{n_i} \quad (2)$$

The corresponding weight is therefore computed as equation (3).

$$W_{i,j} = tf_{i,j} * isf_i \quad (3)$$

### 3. Related Works

Research studies in non-Persian languages domain considers challenges of automatic text summarization and related works to 2009 [3].

Some research studies improve the summarization by using fuzzy logic. As an example, in [12] summary sentences were extracted using fuzzy rules and sets. They extracted the important features for each sentence of the document represented as the vector of features consisting of the following elements: title feature, sentence length, term weight, sentence position, sentence to sentence similarity, proper noun, thematic word and numerical data. They had done experiments with 125 dataset, comparing our summarizer with Microsoft Word 2007 and baseline using precision, recall and f-measure built by ROUGE. The results show that the best average precision, recall and f-measure to summaries produced by the fuzzy method. Certainly, the experimental result is based on fuzzy logic could improve the quality of summary results that based on the general statistic method. In conclusion, they proposed that using combination of fuzzy logic and other learning methods and extract the other features could provide the sentences more important.

In another study an automatic multiple document text summarizers has been presented [13]. Research was first started naively on single document abridgement but recently information is found from various sources about a single topic in different website, journal, newspaper, text book, etc., for which multi-document summarization is required. In [13], automatic multiple documents text summarization task is addressed and different procedure of various researchers are discussed. They compared various techniques that have done for multi-document summarization. Some promising approaches are indicated and particular concentration is dedicated to describe different methods from raw level to similar like human experts, so that in future one can get significant instruction for further analysis.

In addition, there are some studies done in Persian text summarization such as FarsiSum [14] which is inspired by SweSum [15]. This system works based on statistical properties and does not consider text linguistics and special Persian language challenges such as find synonyms, words stemming and etc.

Shamsfard and Karimi [16] propose summarizers based on lexical chain and graph theory that each sub graph show the discrete subject in document. In this method summaries are created based on query and selected of special sub graph, whereas in public summaries the sentences can be selected from all sub graphs.

Furthermore, the Parsina text summarizer employs combination of TF-IDF, graph-based methods and genetic algorithm (GA) [2].

Riahi et. al, [17] propose automatic text summarization using artificial neural networks (ANN) for the weighting system. In this system, importance of parameters are determined using ANN and creates final summary by pruning artificial neural networks.

Bazqandi et. al, [8] have clustered Persian sentences using binary particle swarm optimization where semantic distance of two vectors are used instead of Euclidean distance. The results show that their proposed method assessors, the use of semantic clustering PSO to determine the optimal number of clusters, better accuracy in terms of clustering is compared with other methods.

In this paper we use combine research studies related to Parsina [2], and SA-GA hybrid algorithm [18]. Therefore, the next sub-section describes SA-GA and Parsina text summarizer system, respectively.

#### 3.1. SA-GA Hybrid Algorithm

In this method, we use combination of genetic algorithms (GA) and simulated annealing (SA) to present a new method called SA-GA [18].

In genetic algorithms, chromosomes with three crossover operator, mutation and selection in successive iterations converge to the best solution in the search space. However, while there is variation in population genetic algorithm, convergence to optimality is not guaranteed [18].

Simulated annealing algorithm is an optimization method that finds the optimal locations using random search. In this method, particles with an initial temperature and proceeded to search the solution space are determined. For each particle,  $r_1$  parameter ranges specified in the operating position (Present) by equation (4) where  $\alpha$  is random number between zero and one [18].

$$Present[i+1] = Present[i] + r_1 - r_1 * 2 * \alpha \quad (4)$$

The SA-GA hybrid algorithm employs SA for crossover operation in GA. This method uses the concept of SA for crossover the chromosomes. We have used of real version of Genetic Algorithm (Real-GA). For crossover operation using statistical averages and equation (4) the proposed equation (5) is created.

$$Present[i] = \frac{(Present[i] + Present[i+1])}{2} + r_1 - r_1 * 2 * \alpha \quad (5)$$

Where the Present [i] and Present [i + 1] are two combined parents,  $r_1$  and  $\alpha$  similar to equation (4) are set. Researchers demonstrate performance of these algorithms on complex mathematical functions [18].

#### 3.2. Parsina Text Summarizer

The Parsina text summarizer uses combination of graph-based methods and genetic algorithms [11, 2, 4]. A directed graph is produced after weighting sentences and creates the similarity matrix. The sentences included in the summary based on topic relation factor, cohesion and readability factors are selected. The fitness function in GA, evaluate the sentences. Chromosome length of genetic algorithms is the number of sentences in the summary. The genes per chromosome in population GA represent the number of sentences that there are in summary. This system uses the

Krovetz stemming for noun and verb stemming [19, 20]. Also this system detects the synonym words. The Parsina text summarizer steps are as follows.

Firstly, it detects synonym and stop words with the use of databases. Also performs stemming all words such as nouns and verbs. Secondly, the frequency of words using TF-IDF (equation (1) and (2)), is calculated. The weight of each word according to equation (3) is also calculated. It should be noted that weighting the title words and keywords of the equation (6) is calculated.

$$W_{i,j} = (0.5 + \frac{0.5 * freq_{i,q}}{\max_l freq_{l,q}}) * isf_i \quad (6)$$

Where q represents the user's title or keyword. Third step is the construction of a similarity matrix using equations (7) and (8). In equation (7) similarity of sentence with title is computed. In equation (8) similarity of two sentences together is calculated.

$$sim(s_j, q) = \frac{\sum_{i=1}^t w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} * \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (7)$$

$$sim(s_m, s_n) = \frac{\sum_{i=1}^t w_{i,m} * w_{i,n}}{\sqrt{\sum_{i=1}^t w_{i,m}^2} * \sqrt{\sum_{i=1}^t w_{i,n}^2}} \quad (8)$$

After constructing the similarity matrix, a weighted graph is composed. In fact, the weight of an edge, connecting two vertices, is the similarity of the corresponding sentences (equation (9)).

$$\begin{aligned} \forall (s_i, s_j) \in E, W(s_i, s_j) &= sim(s_i, s_j) \\ \forall i < N : sim(s_i, s_i) &= 0, \forall i, j < N : sim(s_j, s_i) = 0 \end{aligned} \quad (9)$$

A good summary contains sentences that are similar to the text title. Now that the similarities are computed, we can define the Topic Relation Factor (TRF) [2, 4, 11]. A simple method is to consider the average similarity of sentences in the summary, divided by the maximum average. Let TR be the average similarity to title in a summary (s). Using TR, we compute TRF using equation (10). Where max is computed among all possible summaries of length S. To find the max, we should simply average top greater S similarities of all sentences with the topic. TRF shows the similarity of the created summary to the document title. In summaries where sentences are closely related to the title, TRF is close to 1. But in summaries which are constructed by the sentences far from the title, TRF tends to zero.

$$TR_s = \frac{\sum_{s_j \in summary} sim(s_j, q)}{S}, TRF_s = \frac{TR}{\max_{\forall summary} (TR)} \quad (10)$$

Cohesion Factor (CF) is a measure to determine whether

sentences in the summary talk about the same information or not. Suppose C be the average of similarities of all sentences in a summary, it is clear that, C is the average of weights of all edges in the summary sub graph. Where N<sub>s</sub> is the total number of edges in summary sub graph and can easily be computed. Suppose summary nodes are  $S_{s_1}, S_{s_2}, \dots, S_{s_S}$ , and S is the total number of sentences in the summary. Then N<sub>s</sub> is the number of edges from  $S_{s_1}$  to  $S_{s_j}$ ,  $1 < j \leq S$ , plus number of edges from  $S_{s_2}$  to  $S_{s_j}$ ,  $2 < j \leq S$ , therefore:

$$\begin{aligned} C_s &= \frac{\sum_{\forall s_i, s_j \in summary subgraph} W(s_i, s_j)}{N_s} \\ CF_s &= \frac{\log(C * 9 + 1)}{\log(M * 9 + 1)}, M = \max_{i,j \leq N} sim_{i,j} \\ N_s &= (S-1) + (S-2) + \dots = \frac{(S)*(S-1)}{2} \end{aligned} \quad (11)$$

CF should show how summary sentences are close in total, and is defined as equation (11) [2, 4, 11]. M is the maximum weight in the graph, i.e. M is the maximum similarity of sentences. If most of the sentences in the extract talk about same topic, CF grows and on the other hand if they are mostly far from each other, CF tends to 0. Readable extracts are hard to be achieved. A readable document is a document which sentences are highly related to their proceeding sentences. First sentence and second sentence are related to each other with a high similarity, same for second and third sentences, and so on. In fact a readable summary, as we define, is made up of sentences which form a smooth chain of sentences. Suppose the readability of summary s with length S, say R<sub>s</sub>. Therefore, the readability factor of summary s is computed such as equation (12).

$$R_s = \frac{\sum_{0 \leq i < S} W(s_i, s_{i+1})}{\max_{\forall i} R_i}, RF_s = \frac{R_s}{\max_{\forall i} R_i} \quad (12)$$

It should be noted that we have made the assumption that the summary length is fixed and the maximum is computed among all possible summaries of that summary length. Finding this maximum can be done in polynomial time. Suppose the summary length is S, so the goal of finding the most readable summary is equal to finding a path of length S with maximum weight in the document graph. To have such a fitness function we use a function which is the weighted average of three factors such as equation (13) [11, 2, 4]. The fitness function is a flexible one, which means, it has parameters to be adjusted by the user need. One may desire a summary with the highest readability while another may like to have a summary in which sentences are highly related to the topic, and readability is not a matter at all.

$$F = \frac{\alpha * TRF + \beta * CF + \gamma * RF}{\alpha + \beta + \gamma} \quad 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1 - \alpha, \gamma = 1 - \alpha - \beta \quad (13)$$

$\alpha$ ,  $\beta$  and  $\gamma$  are real numbers and are defined by user. TRF; CF; RF are all between 0, 1, so this composition of them results a real number between 0 and 1. Finally, we perform the genetic

algorithm to find the best summary.

#### 4. Phoenix: The Proposed Automatic Text Summarizer System

The proposed system known as Phoenix is the extended version of Parsina text summarizer system [2, 4]. The system aims to improve the quality of summaries produced by using the SA-GA hybrid algorithm. In order to improve the quality of summaries produced by the proposed method the following tasks have been considered: 1) providing a faster method of coding for word stemming krovetz method, 2) Extending the dictionary of words and their synonymous, roots of verbs and nouns to cover more aspects of the text and the increasing use of software in various fields of scientific, artistic, political, or social, and 3) Using SA-GA hybrid algorithm for evaluating the quality of the produced summary.

There is a preprocessing phase in Phoenix system, such as Parsina, to solve the problem bipartite compound words and integration them. The preprocessing phases in phoenix text summarizer system, which display main structure of text, include the following:

- 1 Identifying the boundaries of sentences: In the Phoenix system for identify boundaries of Persian sentences of point, question mark, semicolon are used.
- 2 Removing stop words: words that have no meaning and we cannot collect information about them, such as conjunctions and adverbs that have no role in the concept of sentences.
- 3 Stemming: Finding the root of the derivational words is called stemming. The stemming phases in Phoenix system is similar to Parsina, and uses Krovetz algorithm that have a dictionary of morphological methods for examining the roots [19, 20]. Also the Phoenix system uses an optimized krovetz algorithm. Experimental results show that the stemming method in Parsina is time consuming because of the nested IF statements used in the code. However, In Phoenix system, the Krovetz stemming algorithm is implemented using a function and nested condition statements have been removed. This procedure has improved performance in the stemming phase, and increased its speed compared to Parsina system.

First step in the processing phase of Phoenix system is to compute the effective properties and relation of sentences As well as the TF-IDF weight allocated to sentences. The final score of each sentence are then determined by using an equivalent weight. The best sentences are ranked and selected for the final summary.

The authors have used binary crossover operator in Real-GA for Parsina system, but in the Phoenix system they have used a combination of binary crossover and SA-GA methods in crossover operator. Therefore, the crossover point according to equation (14) for two parents in crossover operator is determined.

$$\text{Crossover Point} = r - r * 2 * \alpha \quad (14)$$

Where  $r$  is the temperature in the SA formula and initial value is a random number between one and the numbers of sentences in the text. Alpha ( $\alpha$ ) is a random number between zero and one. After determining the crossover point at random and using the SA formula, we perform crossover such as binary crossover.

Generally there are three units in Phoenix system named: initializing, scoring and summarizing unit.

Initializing unit includes pre-processing, segmentation and stemming. Scoring unit consists of weighted words (equations (1), (2), (3) and (6)) and constructs the similarity matrix (equations (7), (8) and (9)). The summarizing unit includes the following stages: computing the number of summary sentences, calculating the percentage of Topic Relation Factor (equation (10)), Cohesion (equation (11)), and Readability (equation (12)); and implementing the SA-GA hybrid algorithm.

The SA-GA hybrid algorithm in Phoenix text summarizer system uses SA algorithm in GA crossover operator. In fact the crossover point in chromosome is determined using the concept of SA. The SA-GA algorithm implementation steps are as follows [18]:

- 1 Begin
- 2 Create and initialize a random population.
- 3 Select the best population with elitism.
- 4 Create a new generations using of crossover point (equation (14)) and crossover rate equal with 0.7.
- 5 Perform Mutation with mutation rate 0.1.
- 6 The above steps until you reach the appropriate level of fitness, repeated (Go to 3).
- 7 End

#### 5. Evaluation of the Phoenix Text Summarizer System

The Phoenix summarizer assessment questionnaire		
Question	Phoenix Summarizer	Parsina Summarizer
Category Name		
File Name		
Text Title		
Keywords		
Topic Relation Factor (Percentage)		
Cohesion Factor (Percentage)		
Readability Factor (Percentage)		
Satisfaction with the summaries generated for compress 10%:		
Satisfaction with the summaries generated for compress 25%:		
Satisfaction with the summaries generated for compress 50%:		
Satisfaction with the summaries generated for compress 75%:		
Satisfaction with the summaries generated for compress 90%:		
Mean satisfaction with the summaries produced:		

Figure 1. The Phoenix summarizer assessment questionnaire.

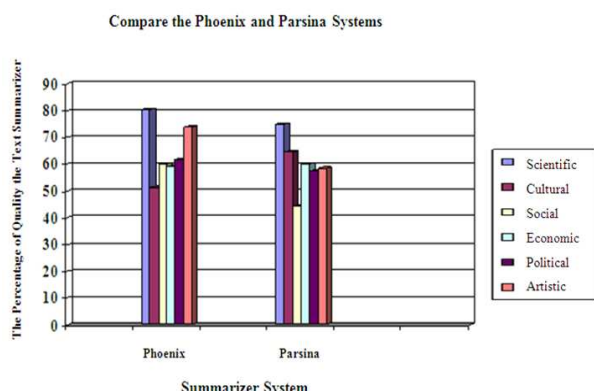
In this section we evaluate the Phoenix text summarizer, by using 6 text categories in various fields such as: scientific, cultural, social, economic, political and artistic. Also the Phoenix and Parsina text summarizers are compared together. Each category includes 5 text files with different lengths, so a total 30 texts were compared. The purpose is to evaluate the quality of text summaries from a human perspective; therefore a team of people are required to evaluate both softwares. We used 10 persons for this work. Each person evaluates the produced summary of Phoenix and Parsina for 30 texts with compress percentages 10%, 25%, 50%, 75% and 90%. We use questionnaire (Figure 1) to register results of both software. Also we considered results in the same conditions. So we used the same title, keywords, topic relation factor, cohesion, readability, and compress percentage for each text in each category.

Results from the questionnaire showed that the average accuracy of the Phoenix automatic text summarization system for each category is as table 1.

**Table 1.** Average accuracy of the Phoenix and Parsina systems.

Category Name	Phoenix	Parsina
Scientific	80.56%	74.78%
Cultural	51.11%	64.44%
Social	60%	44.44%
Economic	58.89%	60%
Political	61.67%	57.22%
Artistic	73.89%	58.33%

So in the Phoenix the average of quality and accuracy is 64.35% while in the Parsina is 59.86%. This proves that the SA-GA hybrid algorithm increases the quality of text summarization. The results show that the Phoenix text summarizer is faster than Parsina text summarizer for short text; but for longer texts is a bit slower than Parsina system with the advantages of having more quality for summary of text compared to Parsina. Figure 2 compares the average quality of text in Phoenix and Parsina text summarizers for various documents show.



**Figure 2.** Comparison chart between Phoenix and Parsina systems.

## 6. Discussion and Conclusion

In this paper we proposed an extended version of Parsina automatic text summarization called as Phoenix. The main

differences between Phoenix and Parsina are: 1) providing a faster way of coding for Krovetz method in word stemming phase, 2) Extending the dictionary of morphological methods for examining the roots have been found, to cover more aspects of the text and improving the applicability of software in various fields such as scientific, cultural, social, economic, political, and artistic fields, 3) Phoenix employs SA-GA hybrid algorithm for determining the sentences of the summary and evaluation of the Summary producer. The results show that the SA-GA hybrid algorithm increases the quality of the text summary in the Phoenix system (64.35%) rather than Parsina system (59.86%).

Future works include using Wordnet for Persian language and develop the Phoenix text summarizer system for more applications and multiple documents.

## References

- [1] Karen Sparck Jones, "Automatic summarizing: factors and directions, in: Advances in Automatic Text Summarization", MIT Press, pp. 1–12, 1999.
- [2] Bahrepour Majid, Mahdipour Elham, Ghalibaf. K. Azadeh, Amiri Malihe, Tahmaseby Aida, Akbarzadeh T. Mohammad Reza, "Automatic Persian Text Summarization", 14th Annual Conference of Computer Society of Iran, Amirkabir University of Technology (Tehran Polytechnic), 2009. (Persian) Available on: [http://www.civilica.com/Paper-ACCSI14-ACCSI14\\_082.html](http://www.civilica.com/Paper-ACCSI14-ACCSI14_082.html)
- [3] Oi Mean Foong, Alan Oxley and Suziah Sulaiman, "Challenges and Trends of Automatic Text Summarization", IJIT, Vol. 1, Issue 1, ISSN: 0976–5972, 2010.
- [4] Mahdipour Elham, Bahrepour Majid, Amiri Malihe, Tahmaseby Aida, "Parsina: Automatic Persian Text Summarizer", Registered software in the development center of information technology and digital media, Register Number: 10.308, Register Date: 2010, Identification Number: 8-00202-000269. Iran, Tehran (Persian)
- [5] Hassel, M., "Resource Lean and Portable Automatic Text Summarization", Stockholm, Sweden, p. 144, 2007.
- [6] Jen-Yuan Yeh, H.-R.K. Wei-Pang Yang, I-Heng Meng, "Text Summarization using a trainable summarizer and latent semantic analysis", Information Processing & Management, Vol. 41, Issue 1, pp: 75–95, 2005.
- [7] Karel Jezek, Josef Stainberger, "Automatic Text Summarization (The state of the art 2007 and new challenges)", Vaclav Snasel (Ed.): Znalosti 2008, pp. 1–12, ISBN 978-80-227-2827-0, FIIT STU Bratislava, Ustav Informatiky a softveroveho inzinierstva, 2008.
- [8] Bazqandi Mahdi, Taday'oun Tabrizi Qamarnaz, "Clustering the sentences based on swarm intelligence", 4<sup>th</sup> Iranian Conference on Electrical and electronic, Islamic Azad University of Gonabad, 2011. (Persian) Available on: [http://www.civilica.com/Paper-ICEEE04-ICEEE04\\_153.html](http://www.civilica.com/Paper-ICEEE04-ICEEE04_153.html)
- [9] Ohtake, K., Okamoto, D., Kodama, M., Masuyama, S., "Yet another summarization system with two modules using empirical knowledge", In Proceeding of NTCIR Workshop2 Meeting, 2001.



- [10] Neto, J., Freitas, A., Kaestner, "Automatic text summarization using machine learning approach", Proc. 16th Brazilian Symp. On Artificial Intelligence (SBIA-2002). Lecture Notes in Artificial Intelligence 2507, pp 205-215, 2002.
- [11] Qazvinian,Vahed.,SharifHassnabadi,Leila.,Halavati, Ramin., "Summarizing Text With a Genetic Algorithm-Based Sentence Extraction", Int. J. Knowledge Management Studies, Vol. 2, No. 4, pp:426-444, 2008, Available on: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.2201&rep=rep1&type=pdf>.
- [12] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan, "Fuzzy Logic Based Method for Improving Text Summarization", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 2, No. 1, 2009.
- [13] Md. Majharul Haque, Suraiya Pervin, and Zerina Begum, "Literature Review of Automatic Multiple Documents Text Summarization", International Journal of Innovation and Applied Studies, ISSN 2028-9324 Vol. 3 No. 1, pp. 121-129, 2013.
- [14] Mazdak,N., Hassel,M., "FarsiSum-a Persian Text Summarizer", Master Thesis, Department of Linguistics, Stockholm University, 2004.
- [15] Dalianis,H., "SweSum- A Text Summarizer for Swedish, Technical Report", TRITANA-p0015, IPLab-174,NADA, KTH, 2000.
- [16] ShamsFard Mehrnoush, Karimi Zohre, "The automatic Persian text summarization system", 12th Annual Conference of Computer Society of Iran, Tehran, 2007. (Persian) Available on: [http://www.civilica.com/Paper-ACCSI12-ACCSI12\\_377.html](http://www.civilica.com/Paper-ACCSI12-ACCSI12_377.html)
- [17] Riahi Noushin, Ghazali Fatemeh, Ghazali Mohammad Ali, "Improved the Persian text summarizer performance using pruning algorithm of neural networks", 1th conference of Line processing and Persian language, Department of Electrical and Computer Engineering, Semnan, Iran, 2011. (Persian) Available on: [http://conf.semnan.ac.ir/uploads/conference\\_khat/Persian/110.pdf](http://conf.semnan.ac.ir/uploads/conference_khat/Persian/110.pdf)
- [18] Mahdipour Elham, Bahrepour Majid, Mohammad Kazemi Farhad, Akbarzadeh T. Mohammad Reza, "A novel method for hybrid of genetic algorithm and simulated annealing", 2nd Joint Congress on Fuzzy and Intelligent Systems, Malek Ashtar University of Technology, Tehran, Iran, 2009. (Persian) Available on: [http://www.civilica.com/Paper-FJCFIS02-FJCFIS02\\_016.html](http://www.civilica.com/Paper-FJCFIS02-FJCFIS02_016.html)
- [19] R. Krovetz, "Viewing morphology as an inference process", Proc.16th ACM SIGIR, 1993.
- [20] Hessami Fard Reza, Ghasem sany Gholamreza, "Design of a stemming algorithm for Persian", 11th Annual Conference of Computer Society of Iran, Tehran, 2006. (Persian) Available on: [http://www.civilica.com/Paper-ACCSI11-ACCSI11\\_066.html](http://www.civilica.com/Paper-ACCSI11-ACCSI11_066.html)