A New Graph-Based Algorithm for Persian Text Summarization

Hassan Shakeri, Saeedeh Gholamrezazadeh, Mohsen Amini Salehi and Fatemeh Ghadamyari

Abstract Nowadays, with increasing volume of electronic text information, the need for production of summary systems becomes essential. Summary systems capture and summarize the most important concepts of the documents and help the user to go through the main points of the text faster and make the processing of information much easier. An important class of such systems is the ones that produce extractive summaries. This summary is produced by selecting most important parts of the document without doing any modification on the main text. One approach for producing this kind of summary is using the graph theory. In this paper a new algorithm based on the graph theory is introduced to select the most important sentences of the document. In this algorithm the nodes and edges will be assigned with different weights and then the final weight of each one will be defined by combining these values. This final weight indicates the importance of the sentence and the probability of appearing this sentence in the final summary. The results show that considering simultaneous different criteria generate a summary which is more similar to human one.

Keywords Summarization · Persian texts · Sentence's graph · Extract · Cohesion

H. Shakeri (✉) · S. Gholamrezazadeh · M. A. Salehi · F. Ghadamyari
Islamic Azad University, Mashhad Branch, Mashhad, Iran
e-mail: shakeri@mshdiau.ac.ir

S. Gholamrezazadeh
e-mail: Saeedeh.motlagh@gmail.com

M. A. Salehi
e-mail: amini@mshdiau.ac.ir

F. Ghadamyari
e-mail: Fatemeh.ghadamyari@gmail.com

James J. (Jong Hyuk) Park et al. (eds.), *Computer Science and Convergence*, Lecture Notes in Electrical Engineering 114, DOI: 10.1007/978-94-007-2792-2_3,

1 Introduction

Nowadays, human faces a large amount of information every day. Significant parts of this information are in text format. Due to this, there is a demand for tools that accelerate the reading and comprehending text documents. Summarization systems enable us to read the most important parts of each document and so increase the speed of reading and comprehending. There are various definitions for summary. Edward Hovy et al. [1] defines the summary as a text that is based on one or more texts; it has the most important information of the main texts and its content is less than half of the main texts. Mani [2], describes the text summarization as a process of finding the main source of information, finding the main important contents and presenting them as a concise text in the predefined template.

Some factors such as the language of input text are so challengeable. For example in Persian texts, multi meaning and multi functional words are one of the challenges in summarizing these texts. In [3] the challenges in Persian texts processing is discussed completely.

There are three main steps for summarizing texts [4]. These steps are *topic identification, interpretation,* and *summary generation.* In topic identification step, the most prominent information in the text is identified. Most of the systems, assign different precedence to different parts of the text (sentence, words, and phrases); then a fuser module mix the scores of each part in order to find the total score for a part. At last, the system presents the N highest score parts in final summary. Several techniques for topic identification have been reported such as methods based on Position, Cue Phrases, word frequency and content counting [1].

Abstract summaries need to go through interpretation step. In This step, related subjects are combined in order to form a general concise content [5] and the additional phrases are omitted. Inferring the topics is difficult; therefore most of the systems generate the extract summary.

In summary generation, the system uses text generation method. This step includes a range of various generation methods from very simple word or phrase printing to more sophisticated phrase merging and sentence generation [9]. In other words, the natural language, which is understandable, by user is generated here.

The summarization systems are categorized based on the type of generated summary. In this paper we focus on extractive summaries. An extractive summary is generated by selecting sentences from the main text to form a summary without any modification of their original wording. Up to now, many different techniques have proposed to select the most important part of the text such as statistical methods which includes Aggregation Similarity Method [6], Location Method [7], Frequency Method [8], TF-Based Query Method [9], linguistic methods which includes Graph Theory, Lexical Chain, WordNet and Clustering. Graph is an appropriate approach for presenting the relation between sentences in a way that the relation between each two sentences can be shown independent the other ones.

Lexical chain uses WordNet in order to identify the relation between words and put them in a chain. Since there is not any WordNet for Persian words, we cannot

use this technique in our system. Clustering technique makes some clusters from sentences and relates them to each other. Clustering method is unable to examine the relation between sentences in a text whereas graph technique provides this relation for us. Considering these reasons, in this paper we propose a graph-based algorithm for summarizing Persian texts.

The structure of this paper is as follows: in Sect. 2, the categorization criteria for summarized text will be discussed. In the third section, we discuss the graph-based approach for summarization and introduce systems which are designed based on it. In the forth section, we introduce our proposed algorithm. In Sect. 5, the experimental results will be analyzed and finally in Sect. 6, the result will be discussed.

2 Related Works

Based on the output summary, the text summarization systems are classified. These categorizations are discussed in [10].

So far, many systems which generate a kind of summary that presented in Fig. 1 are designed. Most of these systems are designed for English texts and other language such as Japanese, Spanish and others. But unfortunately very few efforts have been done for Persian texts. The most important systems for Persian text summarization are introduced here.

FarsiSum: FarsiSum [5, 11] is a text summarization system for Persian newspaper text/HTML. It uses modules implemented in SweSum [14], a Persian stop list in Unicode format and a small set of heuristic rules. The summarization process includes three phases: Tokenizing, Scoring and Keyword extraction. Words in the document are converted from ASCII to UTF-8 and then compared with the words in stop-list.

Automatic Persian Text Summarizer: this system uses a hybrid method to summarize Persian texts automatically. In this system, the below techniques are used to select the sentences which should be presented in final summary: Lexical chains, summarization based on graphs, selecting important sentences based on cue words, number of similar sentences, similarity between sentences and similarity with topic and query. For more details see [12].

Hybrid Farsi text summarization: in [13] a technique based on term co-occurrence and conceptual property of the text is defined. In this method for each two words, the co-occurrence degree is computed. Then lexical chains are created and n top ranked words are selected. After that, a graph is created with words as its nodes. The edges are drawn based on co-occurrence degree between words. The sentence score is computed by summing the gain of all its words and finally n top ranked sentences are selected.

Constructing Graph Algorithm
Input: sentences from the main text, Base node
Output: sentences which make summary
1. For each sentences take part in constructing graph
   1.1. Calculate number of edges connected
   1.2. Calculate total frequency of words
   1.3. Count number of key words
   1.4. Count multiple selection criteria simultaneously
   1.5. Calculate the rate of deviation from base node //base node is calculated earlier by formula number 1 and is identified for this algorithm as input
2. For each two sentences
   2.1. Count shared words
   2.2. Count shared key words
   2.3. Count common English word
   2.4. Count common word with explanation as footer
   2.5. Determine if two sentences are located in one paragraph
3. For each sentences calculate formula number 4

Fig. 1 The proposed summarization algorithm based on graph

3 Applying Graph in Extractive Summaries

Using the graph for displaying the structure of the text will help us to better understand the connection between different parts. [10]. Graph-based algorithms use a ranking algorithm to rank different sections of a text where each section is considered as a node. Ranking algorithms use various criteria in order to sort the nodes based on priority.

The nature of nodes and edges will be defined by the type of text. For example, some sections of the text, words, or sentences can be considered as the nodes. Edges will represent the lexical or semantic connection, or commonalities between the two nodes.

Regardless of the type and characteristics of the text that we want to draw the graph for it, a graph-based ranking algorithm includes the following basic steps:

1. Identify units of text-which can include phrase, word or other units- and to consider them as vertices in the graph.
2. Determine the relationships that these units were related to making and using these relationships to drawn edge between the vertices of the graph. The edges can be directed/undirected and weighted/unweighted.
3. Run the graph ranking algorithm repeatedly until all entities (nodes) are sorted according to priority.
4. Sort the vertices according to their rank.

As it is stated in the fourth step, after specifying the final score for each node, the nodes are ranking based on their final score. Then, depending on compression rates—Compression Rate defines how the main text should be shortened or determines the length of summary text- in the desired text, sentences with the highest score are selected to attend in final summary. LexRank [15] and TextRank

[16] are two of the most important algorithms based on the graph. Following, we examined each of these algorithms briefly.

LexRank: Erkan and Radov [15] designed LexRank in order to summarizing text in multi-document systems. It is assumed that a sentence which is similar with many other sentences in a cluster, is more central (more important) and closer to the subject. In this algorithm, a fully connected and undirected graph is plotted for the sentences of each cluster. If two sentences share similarities, an edge is drawn between them. The cosine similarity is used to calculate the similarity between two sentences. After the calculation of similarity between sentences and construct a graph, we specify the central sentence using graph for each cluster by the following order. They define a degree of centrality for each sentence which is the number of similar sentences to the desired one. The sentence with the highest degree of centrality is the central sentence.

TextRank: TextRank is a graph-based ranking model which is used for all graphs that derived from natural language texts. TextRank is derived from Google page ranking [9] model and is designed for use in single document summarization systems. TextRank is used to extract key words and sentences. A fully connected and undirected graph is used to extract sentences. Each sentence is considered as a vertex (node) in graph. To make an edge between two sentences, a similarity relation is used which is measured as a function of joint concepts. Each edge is also weighted that indicates the importance of relationship. Sentences based on their scores are ranked and the sentences with the highest score are selected [16]. SUMGRAPH [8] and Time stamped Graph [17] are two other summarization systems which are designed based on the graph.

4 Proposed Method

As mentioned earlier, in many of current methods, the most important sentences are selected based on final weight of nodes whereas the weight of edges is solely used to determine the weight of nodes.

The method proposed in this paper tries to involve all existing relations between sentences in determining the most important sentences. Moreover, the importance of sentences independently is considered. In other words, we consider both the importance of each sentence and the importance of relation between sentences. The reason is that if the content of a sentence is not important, it is worthless for the system, no matter how close is the relation with other sentences. The strength of this algorithm is addressing the importance of sentences independently and simultaneously the relations between them.

In this algorithm, a connected and undirected graph is used. We consider undirected graph because it is appropriate well for graphs with weak links [7]. Sentences considered as nodes and relation between them is shown by edges. We consider a weight for nodes and for each edge. Weight of each edge defines the

degree of importance of the relation between two sentences. The following criteria are used for weighting nodes:

- Number of edges connected to node.
- Frequency of words in one sentence.
- Number of keywords in a sentence.
- Having multiple selection criteria simultaneously (criteria based on them sentences are selected from original text to form a graph).
- Rate of deviation from the base node: the base node is the one which is known as the most important and key sentence in the text, and contains the main subject of the text.

To determine the base node, we combine weight of all nodes and work out a value for every node. Then we select the highest value and consider the sentence related to it as the base node. For combining weights we apply the following formula:

$$\mathrm{T_w} = \sum\nolimits_{\mathrm{i}=1}^{5} \mathrm{C_i}\,\mathrm{W_i} \tag{1}$$

where T_w is the overall weight, C_i is coefficient dedicated to ith criteria which indicates the percentage importance of this criterion and W_i is the numeric value of ith criteria which is obtained for a specific sentence. C_i is obtained experimentally and based on a research conducted on the characteristics of Persian texts. After identifying the base node, we calculate the deviation of each sentence from the base node by formula (2).

$$\mathrm{diff} = \frac{\mathrm{N} - \mathrm{I}}{\mathrm{D}} \tag{2}$$

where *diff* is the deviation of each sentence from the base node, N is the number of words in sentences, I is the number of common words with base node and D is the diversity words of the original text. D is calculated by reading text one time and considers each word in its first presence in the text. After calculating the deviation, we reduce the obtained number from the P/D fraction where P is the number of words in base node. Thus, the similarity of each sentence with base node is obtained.

$$\mathrm{S} = \frac{\mathrm{P}}{\mathrm{D}} - \mathrm{diff} \tag{3}$$

Indeed, the sentences which have more similarity with the base node are more related to the topic of the main text. Based on the following criteria, two sentences are related to each other and an edge should be considered between them. These criteria are:

- Number of words shared between two sentences
- Number of keywords shared between two sentences

- Having common English words (this system is designed for Persian texts)
- Having words with common explanation as footer
- Existence of two sentences in a paragraph

After constructing the graph, ten weights are obtained. Five weights are for nodes—former criteria—and five weights are for edges—latter criteria. Then, all these ten weights for all of the nodes and edges are combined using formula (4) and consequently a final weight for each node is calculated. For each node, the weights assigned to itself and weights of edges which is connected to, are combined.

$$T_w = \sum_{j=1}^{10} C_j W_j \tag{4}$$

where T_w is final weight, C_i is coefficient for ith criteria, and W_i is the numerical value for ith criteria which is obtained for a specific node.

The formula 4 is the same formula 1 but the boundaries are changed. In formula 1, we combine just five weights which are obtained by weighting nodes criteria in order to identify the base node whereas the formula 4, combines ten weights obtained by weighting edges criteria addition to weighting nodes, to calculate a final weight for each node.

We assign higher values to the nodes weighting criteria because if one sentence has no importance itself, is not appropriate to attend in the final summary.

Another benefit of the proposed graph to previous algorithms is taking into account the degree of importance for sentences and relationship between them, simultaneously. It enables us to choose the sentences which have main content and are related to the others. This cause the final summary has more cohesion and become more similar with the human one. The following pseudo-code shows the steps of our proposed algorithm.

5 Performance Metrics

To evaluate a text summarization system, two widely used metrics are: Precision and Recall [18]. These two metrics are used just for evaluating extractive summaries.

Recall is the fraction of sentences chosen by the person that were also correctly identified by the system. A person is asked to select sentences that seem to best convey the meaning of the text to be summarized. Then selected sentences automatically by system are evaluated against the human selection.

$$\text{Recall} = \frac{\texttt{system} - \texttt{humanchoicesoverlap}}{\texttt{sentencechosenbyhuman}} \tag{5}$$

And Precision is the fraction of system sentences that were correct.

Table 1 Comparison result between proposed algorithm and FarsiSum

	FarsiSum	Proposed method
Precision	0.37	0.52
Recall	0.52	0.67
F1	0.43	0.58
ROUGE-1	0.43	0.62

$$\text{Precision} = (system - human\ choice\ overlap)/(sentence\ chosen\ by\ system) \quad (6)$$

F1 is a weighted average of the precision, recall and calculated by following formula [19]:

$$F1 = (2 \times (precision \times recall))/((precision + recall)) \quad (7)$$

ROUGE-N [20] is another criteria which is widely used in evaluating summaries. ROUGE-N is calculated as follows:

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{\text{Refrence summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{Refrence summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)} \quad (8)$$

where N stands for the length of the n-gram, gram_n, and $\text{Count}_{\text{match}}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

6 Experimental Results

In order to evaluate the proposed method, we compare the output of this system with FarsiSum [5] system. Evaluation criteria like *Precision*, *Recall*, *standard F1*and *ROUGE-1* are used for comparison.

For this purpose, ten scientific papers about computer technology were summarized by FarsiSum system and the algorithm which is presented in this paper. Also, these papers were summarized by a human expert and we consider this human summary as the reference. Compression rate is set to 50%. The results are listed in Table 1.

As it is shown in Table 1, the precision and recall and ROUGE-1 are improved. We noticed that considering more criteria and specifically taking English words into account is the reason of difference between our approach and approaches applied in FarsiSum. In Persian texts to avoid mistakes and misunderstanding, specialized words are quoted in the original language. We consider these words as clues that express the prominent parts of a professional text. On the other hand, we consider relationship between two sentences in addition to the importance of sentences. This helps generated summary to be more similar with the human one. In fact, for the sake of cohesion and clarity, human summarizer (expert), chooses sentences which are related to each other.

7 Conclusion

In this article we first review the text summarization systems as well as summary classification criteria. Then, we propose a new method based on graph theory to create an extractive summary for Persian texts. The aim of this method is to consider the importance of sentences independently and at the same time the importance of the relationship between them. Thus, the sentences are selected to attend in the final summary contains more important subjects, and also have more contact with other sentences. As a result, we notice that the sentences in summary text have relationship with each other and become closer to the human generated summary.

Evaluation results indicate that the output of proposed method improves precision, recall and ROUGH-1 in comparison with FarsiSum.

This algorithm is a part of text summarization system. In future we plan to fine tune the output of this algorithm. For this goal, we can add some additional processing steps as a post processing step to the system. This step can involve processes such as finding the reference of pronouns in the text and replace them, depends on the genre of the text some sentences can be omitted and reduce the redundancy. Additionally, having richer database improves the accuracy of summary.

References

1. Frankel, David S (2003) Model driven architecture: applying MDA to enterprise computing. OMG Press, Wiley, New York
2. Mani I (2001) Automatic summarization John Benjamin's publishing Co, pp 1–22
3. Shamsfard M (2007) Processing persian texts and its challenges. In: The second workshop on Persian language and computer. pp 172–189. (in Persian)
4. Lin CY, Hovy EH (1997) Identify topic by position. In: Proceedings of 5th conference on applied natural language processing, March 1997
5. Mazdak N (2004) A Persian text summarizer, master thesis, department of linguistics, Stockholm University, Jan 2004
6. Kupiec, Jullian M, Schuetze, Hinrich (2004) System for genre specific summarization of documents, Xerox corporation
7. Rada M (2004) Graph-based ranking algorithms for sentence extraction, applied to text summarization, annual meeting of the ACL 2004, pp 170–173
8. Patil K, Brazdil P (2007) Sumgraph: Text summarization using centrality in the pathfinder network. IADIS Int J Comput Sci Info Sys 2:18–32
9. Wills RS (2006) Google's pagerank: the math behind the search engine
10. Saeedeh G, Mohsen AS, Bahareh G (2009) A comprehensive survey on text summarization systems". CSA 2:462–467
11. Martin H, Nima M (2004) A Persian text summarizer. In: International conference on computational linguistics
12. Zohre K, Mehrnoush S (2007) A system for automatic persian text summarization. In: 12th international CSI computer conference, (in Persian)

13. Azadeh Z, Behrouz M-B, Mohsen S (2008) A new hybrid farsi text summarization technique based on term co-occurrence and conceptual property of the text, In: 9th ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing
14. Dalianis H (2000) SweSum—A text summarizer for Swedish, Technical report, TRITA-NA-P0015, IPLab-174, NADA, KTH, Oct 2000
15. Erkan G, Radev DR (2004) LexRank: graph-based centrality as salience in text summarization, J Artif Intell Res 22, pp 457–459
16. Rada M, Tarau P (2004) TextRank: bringing order into texts. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP 2004)
17. Lin Z (2006–07) Graph-Based methods for automatic text summarization, Ph.D. thesis, school of computing National University of Singapore 2006–07
18. Nenkova A (2006) summarization evaluation for text and speech: issues and approaches, Stanford University
19. Norshuhani Z, Arian G (2010) A hybrid approach for malay text summarizer, The 3rd international multi-conference on engineering and technological innovation 2010
20. Lin C (2004) Rouge: a package for automatic evaluation of summaries. In: proceedings of the workshop on text summarization branches out, 42nd annual meeting of the association for computational linguistics. 25–26 July, Barcelona, Spain, pp 74–81