Global Journal of Computer Sciences: Theory and Research

Volume 10, Issue 2, (2020) 48-56 www.gjcs.eu

Single document summarisation based on Grey Wolf Optimisation

Moein Salimi Sartakhti*, Amirkabir University, Department of Computer Engineer, Tehran, Iran

Ahmad Yoosofan, University of Kashan, Electrical and Computer Engineering Department, Kashan, Iran

Ali Asghar Fatehi, University of Kashan, Electrical and Computer Engineering Department, Kashan, Iran

Ali Rahimid, VIT University, School of Social Sciences and Languages, Vellore 632014, India

Suggested Citation:

Sartakhti, S. M, Yoosofan, A., Fatehi, A. & Rahimid, A. (2020) Single document summarisation based on Grey Wolf Optimisation. *Global Journal of Computer Sciences: Theory and Research*. *10*(2), 48-030. https://doi.org/10.18844/gjcs.v10i2.5807

Received June 30, 2020; revised from August 15, 2020; accepted from October 17, 2020.

Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, Cyprus.

Abstract

The amazing growth of online services has caused an information explosion issue. Text summarisation is condensing the text into a small version and preserving its overall concept. Text summarisation is an important way to extract significant information from documents and offer that information to the user in an abbreviated form while preserving its major content. For human beings, it is very difficult to summarise large documents. To do this, this paper uses some sentence features and word features. These features assign scores to all the sentences. In this paper, we combine these features by Grey Wolf Optimiser (GWO). Optimisation of features gives better results than using individual features. This is the first attempt to show the performance of GWO for Persian text summarisation. The proposed method is compared with the genetic algorithm and the evolutionary strategy. The results show that our model will be useful in this research area.

Keywords: Text summarisation, genetic algorithm, sentence, score function, evolutionary strategy.

* ADDRESS FOR CORRESPONDENCE: **Moein Salimi Sartakhti,** Department of Computer Engineer, Amirkabir University, Tehran, Iran. *E-mail address*: artakhti.salimi@gmail.com

1. Introduction

Because of the fast growth of the communication instruments and social media, massive amounts of textual data have been produced and saved in different locations, like e-government and e-library in different devices. Nevertheless, these valuable data that are in text form have not been well organised and properly used. Hence, the ongoing growth of available online text documents causes research and application of text summarisation to be very significant. One of the most important methods to manage rising amounts of textual data is automatic text summarisation (ATS). Two kinds of summarisation exist: extractive and abstractive. The extractive summarisation tries to identify the most important sentences of text, while in the abstractive summarisation method, new sentences from given text will be produced. Because of the possibility and feasibility of extractive summarisation method, the furthermost effort in the literature is about extractive summarisation. These studies commonly try to produce a fair score for each sentence separately, and then synthesise them using proportionate weights. There are two methods for finding appropriate weights of features: manually and automatic. Automatic methods are based on supervised learning and they use a heuristic algorithm, e.g., genetic algorithm (GA) [1], ABC algorithm [2] and particle swarm optimisation [3]. Manual methods [4] choose weights of features using experienced and skilled opinions. Unlike manual methods, in automatic methods, this paper deals with training and testing phases of a given corpus. In addition, it is clear that for each domain, these weights are different. After selecting the related weights for the features of sentences, ATS can rank all of the sentences based on their score function's values. Finally, we can extract the highest scored sentences to create summary documents.

This article has leveraged from the Grey Wolf Optimiser (GWO) [5] as algorithm that can combine some sentences' features. To uncover the robustness of the proposed method, there are many methods that can be applied and then compared to GWO. This work uses GA and evolutionary strategy (ES) for comparison. Evaluation of these methods together can show that our method produces better results than other approaches.

2. Literature review for AT

There are different methods for summarisation; some methods go back to even 50 years ago. In recent years, with the growth of technology, ATS has still been a favourite method. This section reviews some of the literature on extractive ATS and GWO. The reviews are presented in this segment.

To generate the extractive summary document, one must select the most important sentences of the text so that selected sentences represent the given text in the best form. Many researches analyse fundamental and semantic features of the text. Some of these features include the following:

- Term frequency;
- Title feature;
- *N*-gram words;
- Sentence length;
- Location;
- Centrality of sentences.

The study [6] has created the first summarisation system based on word frequency occurrence in a given text. Some of the other studies use different shallow features [7], [8]. In addition, some works try to represent the documents with semantic sentence features via probabilistic latent semantic analysis [9], latent semantic analysis (LSA) [10] and non-negative matrix factorisation [11], which explore the relationships between a set of sentences and words by generating a set of subjects related to sentences and words.

2.1. Literature review for GWO

The idea of the GWO algorithm has originated from headship hierarchy and hunting manner of grey wolves that live in nature. The GWO algorithm uses some kind of grey wolves including alpha, beta, delta and omega. Actually, there are three steps: searching for prey, baiting prey and attacking prey. Every wolf has specifications that, in iterations, try to approximate goal specifications. After the end of each iteration, the closest wolf to the goal is called 'alpha' and the rest of the wolves, based on the degree of their proximity to the goal specifications, are called beta, delta and omega. Then these four wolves' specifications are saved for next iteration.

2.1.1. Sentence features

Each sentence is analysed with nine feature. Actually, each sentence is displayed with a nine-dimensional vector. The features are shown in Table 1.

At first, this work did some pre-processing steps on the texts like removing stop words.

Table 1. Description of the features

Sentence score	Name
f_1	Sentence location
f_2	Distributional features
f_3	Similarity to title sentence
f_4	Sentence length
f_5	Term frequency
f_6	Word sentence
f_7	Positive keywords
f_8	LSA based
f_9	Centrality

2.1.1.1. f_1 – sentence location feature: This feature indicates that sentences at the beginning of a paragraph are more important than the rest of the sentences of that paragraph. The first sentences of a paragraph mostly express the topic of that paragraph. The score of each sentence is calculated according to Equation (1) as follows:

$$Score_{f_1}(S_i) = \frac{N-p_i}{N} \qquad (1)$$

where $\forall S_i \in$ d (document) and P_i is the position of the i^{th} sentence and *N* is the total number of sentences of the document.

2.1.1.2. f_2 – distributional feature: One way to find the importance of the term is calculating the compactness of its distribution. The study by [12] uses three features: $ComPact_{\mathrm{PartNum}}$, $ComPact_{\mathrm{FLDist}}$, and $ComPact_{\mathrm{Pos\ Var}}$.

Assume that the array of term t_k is array (t_k, d) = (c_1, c_2, … , c_N), where the frequency of the term t_k in S_N is c_N.

$ComPact_{\mathrm{PartNum}}$ determines the number of sentences where a word appears on them. So it is possible to find out that a word is compact or not. If a word appears in different sentences of a document, it is less compact. This feature is computed as shown in Equation. (2) as follows:

$$ComPact_{\mathrm{PartNum}}(t_k, d) = \sum_{i=1}^{N} c_i > 0\ ?\ 1:0 \qquad (2)$$

The phrase $c_i > 0\ ?\ 1:0$ means that if c_i is greater than zero then c_i = 1 otherwise c_i = 0.

$ComPact_{\text{FLDist}}$ feature tries to calculate the distance between the first and last appearance of a word. If the distance between the first and last appearance is long, then the term enjoys less compact.

$$\begin{aligned} &ComPact_{\text{FLDist}}(t_k, d) = \text{Last}_{App}(t_k, d) - \text{First}_{App}(t_k, d) \\ &First_{App}(t_k, d) = min_{i\in\{1\ldots N\}} c_i > 0 \text{ ? i:N} \\ &Last_{App}(t_k, d) = max_{i\in\{1\ldots N\}} c_i > 0 \text{ ? i:}{-1} \end{aligned} \quad (3)$$

The phrase $c_i > 0 \text{ ? i: N}$ means that if c_i is greater than zero, then $First_{App}(t_k, d)$ = *i* otherwise $First_{App}(t_k, d)$ = *N*. In addition, the phrase $c_i > 0 \text{ ? } 1{:}-1$ means that if c_i is greater than zero, then $Last_{App}(t_k, d)$ = 1 otherwise $Last_{App}(t_k, d)$ = −1.

$ComPact_{PosVar}$ determines the variance of position of all appearances. This measure is computed as shown in Equation (4) as follows:

$$\begin{aligned} &ComPact_{PosVar}(t_k, d) = \frac{\sum_{i=1}^{m} c_i * |i - centroid(t_k, d)|}{count(t_k, d)} \\ &count(t_k, d) = \sum_{i=1}^{N} c_i \\ &centroid(t_k, d) = \frac{\sum_{i=1}^{N} c_i * i}{count(t_k, d)} \end{aligned} \quad (4)$$

where m_i is the total number of different terms in S_i sentence.

According to the study, the distributional feature is computed based on Equation. (5) as follows:

$$Score_{f_2}(S_i) = \sum_{k=1}^{N}(ComPact_{PartNum}(t_k, d) + ComPact_{FLDist}(t_k, d) + ComPact_{PosVar}(t_k, d)) \quad (5)$$

2.1.1.3. f_3 – similarity to title feature: According to this feature, the score of sentences is calculated based on its similarity to the title in the document. The score of each sentence is calculated according to Equation (6) as follows:

$$Score_{f_3}(S_i) = Cosine_{similarity}(S_i, \text{title}) \quad (6)$$

2.1.1.4. f_4 – sentence length feature: According to this feature, it is expected that longer sentences include more knowledge. For a sentence S_i in a document *d*, the score of each sentences is calculated according to Equation (7) as follows:

$$Score_{f_4}(S_i) = \text{total number of terms in } S_i \quad (7)$$

2.1.1.5. f_5 – term frequency feature: This feature assumes that the value of the term depends on number of occurrences in the document [13]. The score of each sentence is calculated according to Equation (8) as follows:

$$Score_{f_5}(S_i) = \sum_{k=1}^{m_i} tf(d, t_k) \quad (8)$$

where m_i is the total of different terms in S_i and *tf* (d, t_k) is the number of times term t_k occurs in document *d*.

2.1.1.6. f_6 – word sentence score feature: This feature directly depends on the frequency of a term and inverse sentence frequency (TF-ISF) of t_n in S_i (*I* = 1,...,*N*), while *N* is total number of sentences in the document [14].

The TF-ISF score of t_k is S_i is calculated as shown in Equation (9) as follows:

$$\text{TF–ISF}(S_i, t_k) = \text{tf}(S_i, t_k) * [1 - \frac{log(sf(t_k)+1)}{log(N+1)}] \quad (9)$$

where *tf*(S_i, t_k) is the number of times t_k occurs in S_i and *sf*(t_k) is the number of sentences including the term t_k.

For a sentence S_i in the document *d*, this feature is calculated according to Equation (10) as follows:

$$Score_{f_6}(S_i) = 0.1 + \frac{\sum_{k=1}^{m_i} TF_s - ISF(S_i,t_k)(d,t_k)}{HTFS} \qquad (10)$$

where HTFS is the highest (TF-ISF) summation among all of the sentences in the document. In addition, m_i is the total number of different terms in S_i and LS is summary length.

2.1.1.7. f_7 – positive word feature: In the Persian language, a positive word includes 'خلاصه' (kholaeseh: summary), 'پایان' (payan: the end), 'نهایتا' (nahayatan: finally), (banabarin: hence) 'بنابراین' etc. that are used in important sentences. This feature is calculated according to Equation. (11) as follows:

$$Score_{f_7}(S_i) = \text{total number of positive words in } S_i \qquad (11)$$

2.1.1.8. f_8 – LSA-based feature: The goal of this feature that is based on [10]is to select sentences that are related to all significant subjects of the document. At first, in this feature must is carried out in the singular value decomposition procedure on a term-sentence matrix of the document. Therefore, singular vector matrix V^T and the diagonal matrix Σ are acquired. Now, vector space *B* is created in line with Equation (12) as follows:

$$B = \Sigma^2 V^T \qquad (12)$$

Therefore, each sentence gives a score by using Equation (13) as follows:

$$Score_{f_8}(S_i) = \sqrt{\sum_{m=1}^{r} b_{mi}^2} \qquad (13)$$

where *r* is the number of dimensions in the new latent space and b_{mi} values are the value of matrix *B*. In this study, the best value for *r* is equal to the length of all of different words divided by 10. This value is acquired in the experiment.

2.1.1.9. f_9 – centrality: This feature combines the similarity, the common friend and the common *n*-grams among the intended sentence and all the other sentences. Then for the normalisation, this feature is divided into *N*−1 in which *N* is the number of sentences in the document. This sentence feature is expressed as in Equation (14) as follows:

$$Score_{f_9}(S_i) = \frac{\sum_j^{N-1} sim(S_i,S_j)}{N-1} + W\ \frac{\sum_j^{N-1} n-friends(S_i,S_j)}{N-1} + \frac{\sum_j^{N-1} n-grams(S_i,S_j)}{N-1} \qquad (14)$$
$$|\, i \neq j \ sim\ (S_i, S_j)) > \alpha$$

where S_i is the intended sentence and S_j is the other sentences in a given document. In addition, *N* is the number of sentences in the document and *α* is the similarity threshold that is obtained experimental. In our study, using experimental, the similarity threshold is taken as 0.03 and the best *n*-gram is taken as bigram.

2.2. Combining sentences features with GWO

At the first step, all of scores of sentence features are normalised to the range (0,1). Then each sentence was given a score that was achieved from combination all of features. The combination feature is calculated as shown in Equation (15) as follows:

$$\text{Score } (S_k) = \sum_{i=1}^{9} w_i\, f_i \qquad (15)$$

where w_i determines the weights of features. In this segment, based on [5] paper, we express what is the idea of GWO and how the GWO acquires weight.

The GWO has a number of parameters that are initialised, they are as follows:

- The number of dimensions of each wolf: This parameter, based on the problem, can accept any number. In our study, because there are five main classes' weights and 12 weights for sentence features, the number of dimensions is 17.
- The number of wolves: This parameter determines how many wolves' algorithms are needed to reach the goal. This parameter can be determined through the experiment.
- The number of attacks: This parameter determines how many times the GWO attacks. This parameter can be determined through the experiment.

In our study, the number of wolves is equal to 5 and the number of attacks is equal to 500.

3. Corpus

In order to find the efficiency of the GWO system, this paper has used a Persian corpus that has both its document and its abstract. This corpus includes 209 documents. Also, the data set has been divided to two sets. One of them called 'train' includes 168 documents and the other one that is called 'test' includes 68 documents. This data set is human-generated abstractive summary.

Table 2. Attributes of the Persian data corpus

Attributes of the data corpus	Train	Test
Number of docs	141	68
Total number of sentences	16353	9674
Min sentences/doc	5	4
Max sentences/doc	619	651
Total number of words	394770	226948
Min Words/doc	151	102
Max Words/doc	16341	12831

4. Results

Our proposed method used the *F*-measure metric to show degree of similarity between summarised text by human and summarised text by system. *F*-measure uses precision (*P*) and recall (*R*). Suppose that *S* is the system-generated summary and *T* is the reference summary, then these measures are defined as follows:

$$P = \frac{|S \cap T|}{|S|},\ R = \frac{|S \cap T|}{|T|},\ F = \frac{2PR}{R+P} \qquad (16)$$

At the first, we calculate F-measure for the training data set. The results are shown in Table 3. As shown in Table 3, GWO has the best F-score. Based on Table 3, the best sentence feature for our data set is centrality feature (f_9) where its value is 0.4384. The value of LSA-based feature (f_8) is very close to the value of centrality feature (f_9). Meanwhile, it is clear that using the combining sentence feature by GWO algorithm increases the performance of the system. The value of GWO is 0.5891. f_{GWO} is the highest value. Based on our data set, the weakest sentence feature is positive key words (f_7) and its value is 0.2589. It is clear that GWO performs better than ES and GA. GWO, GA and ES try to learn weights on training data set.

Weights of GA and GWO are shown in Table 4.

Table 3. Performance results of each feature on training the Persian data set

Sentence features	name
f_1	0.3402
f_2	0.3911
f_3	0.3693
f_4	0.3048
f_5	0.3675
f_6	0.3263
f_7	0.2589
f_8	0.4237
f_9	0.4384
f_{GWO}	0.5891
f_{GA}	0.5470
f_{ES}	0.5711

Table 4. Weights

Sentence features	Value
f_1	0.2975
f_2	0.3128
f_3	0.3696
f_4	0.2509
f_5	0.4077
f_6	0.2961
f_7	0.2684
f_8	0.4328
f_9	0.4392
f_{GWO}	0.4617
f_{GA}	0.4403
f_{ES}	0.4515

Table 5. Performance results of each feature on testing the Persian data set

Weights	GWO	GA	ES
w_1	0.1716	0.2172	0.1018
w_2	0.2112	0.2049	0.2401
w_3	0.1701	0.1254	0.1359
w_4	0.1001	0.1598	0.1496
w_5	0.1518	0.2590	0.3914
w_6	0.1038	0.1272	0.1790
w_7	0.0410	0.1066	0.0280
w_8	0.3319	0.2809	0.4012
w_9	0.3492	0.2421	0.1095

The results of each sentence feature, GA and GWO, on testing the data set are shown in Table 5. We used weights of GWO and GA in Table 4 to extract important sentences in testing data set.

The results in Table 5 show the best sentence feature in testing data set is centrality feature (f_9), where its value is 0.4392 and then value of LSA-based feature (f_8) is highest with 0.4328. It can also be depicted from Table 5 that GWO (0.4617) performs better than GA (0.4403) and ES (4515). Therefore, the proposed method has better performance rather than GA.

5. Conclusion

In brief, the contribution of this article was about using a new optimiser algorithm in ATS. This paper compared our method with GA and showed that GWO could be more valid, reliable and effective than GWO at least in our data set.

Future proposed research should be carried out as follows:

- Using text clustering and GWO simultaneously;
- Using other sentence features;
- Using ensemble algorithm includes GWO, GA etc.;
- Detecting the type of text from weights of GWO! Is it possible?
- Using another data set.

References

[1] E. Adamcsek, “The analytic hierarchy process and its generalizations,” Ph.D. dissertation, Eotvos Lorand Univ., Budapest, Hungary, 2008.

[2] J. M. Sanchez-Gomez, M. A. Vega-Rodr\'{\i}guez, and C. J. P\'{e}rez, “Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach,” *Knowl.-Based Syst.*, vol. 159, pp. 1--8, Nov. 2018.

[3] S. Mandal, G. K. Singh, and A. Pal, “PSO-based text summarization approach using sentiment analysis,” in *Computing, Communication and Signal Processing*. Springer, 2019, pp. 845--854.

[4] R. M. Alguliev, R. M. Aliguliyev, M. S. Hajirahimova, and C. A. Mehdiyev, “MCMR: Maximum coverage and minimum redundant text summarization model,” *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14514--14522, 2011.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46--61, Mar. 2014.

[6] H. P. Luhn, "The automatic creation of literature abstracts,'' *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159--165, 1958.

[7] A. G\"{u}ran, N. G. Bayazit, and M. Z. G\"{u}rb\"{u}z, "Efficient feature integration with Wikipedia-based semantic feature extraction for Turkish text summarization,'' *Turkish J. Electr. Eng. Comput. Sci.*, vol. 21, no. 5, pp. 1411--1425, 2013.

[8] \"{O}. Co\c{s}gun, Y. Ekinci, and S. Yan{\i}k, "Fuzzy rule-based demand forecasting for dynamic pricing of a maritime company," *Knowl.-Based Syst.*, vol. 70, pp. 88--96, Nov. 2014.

[9] H. Bhandari, M. Shimbo, T. Ito, and Y. Matsumoto, "Generic text summarization using probabilistic latent semantic indexing," in *Proc. 3rd Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2008, pp. 1--8.

[10] M. Anjaneyulu, S. Sarma, P. V. P. Reddy, K. P. Chander, and S. Nagaprasad, "Topic oriented multi-document summarization using LSA, syntactic and semantic features,'' in *Proc. Int. Conf. Innov. Comput. Commun.* Springer, 2019, pp. 487--502.

[11] A. Sawwalakhe, N. Wanjari, S. Paliwal, S. Katare, and V. Malve, "A survey on various techniques for multi-document summarization,'' Tech. Rep., 2019.

[12] J. Steinberger, "Text summarization within the LSA framework," Ph.D. dissertation, Univ. West Bohemia Pilsen, Pilsen, Czech Republic, 2007.

[13] M. Berker and T. G\"{u}ng\"{o}r, "Using genetic algorithms with lexical chains for automatic text summarization," Bo\u{g}azi\c{c}i Univ., Istanbul, Turkey, Tech. Rep., 2011.

[14] M. S. Binwahlan, N. Salim, and L. Suanmali, "Fuzzy swarm diversity hybrid model for text summarization," *Inf. Process. Manage.*, vol. 46, no. 5, pp. 571--588, 2010.