






Multi-document Text Summarization Based on Genetic Algorithm and the Relevance of Sentence Features

Verónica Neri-Mendoza^(✉), Yulia Ledeneva^(✉) ,
René Arnulfo García-Hernández , and Ángel Hernández-Castañeda 

Autonomous University of the State of Mexico, Instituto Literario No. 100, 50000 Toluca,
Mexico

vnerim001@alumno.uaemex.mx,
{ynledeneva, reagarciah, anhernandezc}@uaemex.mx

Abstract. Document Text Summarization aims to create a short and condensed version from the original document, which transmits the main idea of the document in a few words. We formulated extractive multi-document text summarization as a combinatorial optimization problem. In which we used sentence features to select the most important content. We conduct experiments on Document Understanding Conference (DUC01) dataset using the ROUGE toolkit. Our experiments demonstrate that the proposed method contributes significant improvements over the state-of-the-art methods and heuristics.

Keywords: Multi-document · Text · Summarization · Genetic algorithm · Sentence features · Optimization

1 Introduction

The growth of the Internet involves that documents spread swiftly. Thus, the users get engulfed in many documents, wondering where to access them. In this context, Document Text Summarization (DTS) appears as a viable solution because it aims to generate a condensed version of documents and convey relevant information to the reader. Therefore, users can save time through summaries instead of reading the whole set document to capture the main idea [1, 2]. Due to this situation, researchers in Natural Language Processing are focused on the text summarization task [1]. Optimization-based approaches have been gaining importance because of the excellent performance obtained due to these being effective to get an optimal solution for huge and varied spaces [3–5]. These helps recognize the appropriate sentences to include in a summary in the DTS context.

A domain that has been the object of study in state-of-the-art is news. The different news sources that report on a particular event contain common components that construct the main facts. Thus, DTS from multiple news articles is a valuable field of study since the number of online publications is overgrowing. This is essential to satisfy the information need of various users. For this reason, multiple datasets have been developed, such as

DUC [6], TAC (Text Analysis Conference) [7], Multi-News [8], CNN [9], among others, to evaluate the effectiveness of state-of-the-art methods.

There are three approaches to generating text summaries in the literature: extractive, abstractive, and hybrid.

Extractive Text Summarization. Proposed systems based on this approach create summaries by assigning weights to sentences according to linguistic and statistical features, then selecting the sentences with better weight by combining them. These methods generally contain two significant components: ranking and selection of sentences. In addition, extractive summarization methods ensure the generated summaries are semantically similar to the original documents [3, 10].

Abstractive Text Summarization. This approach allows the proposed methods to create summaries using new corpus words and sentences. The processing of abstractive summarization is like the human generation of summaries. However, it requires sophisticated natural language understanding and generation techniques, such as paraphrasing and sentence fusion [10, 11].

Hybrid Text Summarization. This approach combines the advantages of extractive and abstractive methods to process the input texts. The hybrid approach processes data in two steps: The first step is to reduce the input length of documents to create a selective summary. Afterward, the selective summary is used by an abstractive method to construct a final summary [3].

Depending on the number of documents, summarization can be classified into two tasks: Single-Document Text Summarization, which composes a summary from one document, and Multi-Document Text Summarization (MDTS), which produces a summary from a collection of documents about a particular topic [1, 3, 12].

We formulated MDTS as a combinatorial optimization problem, which we address through a Genetic Algorithm (GA). The GA does not require external resources, working in an unsupervised way. Moreover, we hypothesize is that both sentence position and coverage provide essential information to distinguish relevant sentences from documents to create news summaries. Additionally, we have tested the proposed method by generating summaries of 50, 100, and 200 words on the DUC01 dataset.

The rest of the paper is organized as follows: Sect. 2 presents the related work. Then, Sect. 3 describes the proposed summarization method. In Sect. 4, we show experimental results. Finally, the conclusions of this paper are drawn in Sect. 5.

2 Related Works

In the literature, the DTS has been tackled through many techniques, such as supervised-based methods convert the summarization task into supervised classification problem. Generally, these methods learn by training to classify sentences, indicating whether a sentence is included in the summary. State-of-the-art approaches usually use word embeddings for representing the contextual meaning of sentences. Nevertheless, proposed

methods require a corpus manually staggered [3, 8]. On the other hand, unsupervised-based methods generally assign a score to each sentence of each document, describing the relevance of sentences in the text. Therefore, sentences with the highest values will be part of the extractive summary. [3, 5, 13]. In this approach, four steps have been identified to generate a summary: Term selection, term weighting, sentence weighting, and sentence selection [13]. For the last step, various textual features have been developed [13]. Some of them are presented in Table 1:

Table 1. Unsupervised features.

Feature	Description
Similarity with the title	This feature assigns the most important to the sentences that include words in the title [13, 14]
Similarity with other sentences	Given a sentence called the central sentence, a score is given to the other sentences of the document which contain overlapping words [3]
Sentence length	It assumes that the length of a sentence can indicate whether it is relevant to the final summary. Shorter sentences are usually not included [13, 14]
Redundancy reduction	Redundant or duplicate information in the generated summary is expected to be minimized [3]
Sentence position	The idea is that the first sentences indicate a relevant sentence [3, 14]
Coverage	This feature is based on the idea that information provided in the original documents should be included in the generated summary [3, 13]

3 Proposed Method

In MDTS, the search space is more extensive than in Single DTS, making it more challenging to select the most important sentences. In this context, MDTS can be determined as an optimization problem. The documents from the collection are considered a set of sentences, and the aim is to choose an optimal subset from sentences under a length constraint. Previous works [15, 16] have proposed the GA as an alternative for the MDTS to select an optimal combinatorial subset of sentences, obtaining competitive results compared to other state-of-the-art alternatives. However, we intend to improve its performance. Therefore, we have sought to enhance the GA exploration by increasing the size of the population. The population size is one of the essential factors that affect performance [4, 17]. In general, small population sizes might lead to premature convergence and yield substandard solutions [18].

3.1 Pre-processing

In this step, the documents of each collection were ordered chronologically. Then, the sentences of documents were hierarchically ordered according to appearance in the text to create a meta-document, which contains all collection sentences. Afterward, the text of the meta-document was separated into sentences. Finally, a lexical analysis was applied to separate sentences into words [5].

3.2 Text Modeling

After preprocessing the text, it is necessary to model it. This stage aims to predict the probability of natural word sequences. The simplest and most successful form for text modeling is the n-gram, which is a text representation model that constructs contiguous subsequences of consecutive words from a given text [5].

3.3 Weighting and Selection of Sentences

Sentence weighting and selection of sentences usually worked together [13]. While the first one assigns a degree of relevance for each sentence, the second one chooses the most appropriate sentences to generate extractive summaries. However, it involves a vast search space that requires to be addressed by optimization. In view of this, we propose the following GA to select the most important sentences:

Encoding: *The binary encoding* was used, where each sentence of the meta-document represents a gene. The values 1 and 0 define if a sentence will be selected in the final summary [5, 13, 16].

Generation of Population: The initial population was randomly generated. On the other hand, the population of the next generations is generated from the selection stage. The search process concludes when a termination criterion is met. Otherwise, a new generation will be produced, and the search process will continue [5, 13].

Size of the Initial Population: The size of the population was determined according to the number of sentences from the meta-document [4, 5, 13].

Selection Operator: The selection of individuals is performed through the *roulette operator*, which selects individuals of a population according to their fitness to choose individuals with a higher value. Each individual is assigned to a proportional part of the roulette according to its fitness in this operator. Finally, the selection of parents is performed, which are needed to create the next generation, and each selected individual is copied into the parent population [5].

Crossover Operator: Crossover is a genetic operator that combines two parents to produce one or two descendants. The idea underlying crossover is that the new individual can be better than its parents if it takes the best characteristics of each parent. *Crosses with priority over common genes:* This crossover operator was designed to generate summaries, where each individual represents a selection of sentences. Of the selected parents, only one gene is randomly selected (with value 1) that will belong to the descendant to fulfill the number of words [4, 5].

Mutation Operator: We used the *flipping operator*, which consist of changing the value of each gene, inverting from 1 to 0 or vice versa [5, 13]. First, the mutation is performed considering the genes with a value of 1 and later considering the genes with a value of 0. Afterward, it is verified that the established number of words is fulfilled. If it is not fulfilled, another gene with a value of 0 will be inverted, and this process will continue until the specified minimum number of words is satisfied.

The Fitness Function: It was calculated by employing the concept of the slope of the line [4, 5, 16]. The slope defines the importance of sentences. The main idea is to consider the first sentence with the importance X_n , the second with the significance of $X_n - 1$. In a text with n sentences, if the sentence i is selected for the summary, its relevance is defined as $t(i - x) + x$, where $x = 1 + (n - 1)/2$ and t is the slope to be discovered. The formula to calculate the importance of the sentence position is in Eq. 1:

$$\text{Sentence importance} = \frac{\sum_{|c_i|}^n = 1^{t(i-x)+x}}{\sum_{j=1}^k t(j-x) + 1}, x = 1 + \frac{(n-1)}{2} \quad (1)$$

where k is the number of selected sentences. On the other hand, the content coverage to retrieve all aspects from meta-document was calculated by the summation of the frequencies of the n-grams that the summary weighs. (*Precision_Recall*) was calculated via the sum of the frequencies of the n-grams considered in the original text divided by sum of the frequencies of the different n-grams of summary (see Eq. 2).

$$\text{Precision_Recall} = \frac{\sum \text{Original text frequency}}{\sum \text{Frequency Summary}} \quad (2)$$

Finally, to obtain the value of the fitness function, the following formula was applied, which is multiplied by 1000 (see Eq. 3).

$$FA = \text{Precision_Recall} * \text{Sentence Importance} * 1000 \quad (3)$$

Stop Condition: For this operator, we have used the *number of generations* as a stop condition.

4 Experimental and Results

4.1 Dataset

To empirically evaluate the results of the proposed method, we use the DUC01 dataset, is an open benchmark for generic automatic summarization evaluation, which is in the English language; it is composed of 309 documents split into 30 collections, which we tested with the lengths of 50, 100, and 200 words. We choose this dataset because the gold standards summaries provided in it were typed like an abstractive approach. It allowed us to measure how competitive the proposed extractive unsupervised method can be about summaries made using paraphrases, words, and sentences that do not belong to source documents.

4.2 Evaluation Measures

ROUGE (Recall-Oriented Understudy for Gisting Evaluation). It involves measures to automatically establish the quality of a summary created by a proposed method by contrasting it to other ideal summaries created by humans, called gold standard summaries [20]. These measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans [21].

4.3 Parameter Selection

We perform tests with different parameters such as tournament and roulette selection operator, HUX crossover operator, crossover with priority on common genes, double inversion mutation, with different crossover and mutation probabilities, respectively. Also conducted our tests with varying population sizes; we multiplied the number of sentences of the meta-document from 2 and 15 to determine the best possible population size to improve the GA exploration. Per our empirical results, we conclude that good traits spread through the population for the different summaries lengths (50, 100, and 200 words) by multiplying the number of sentences from the meta-document by 9 and throughout 150 generations. Favoring the selection as parents of individuals with greater fitness value by roulette operator. Moreover, we tested n-grams of sizes from 1 to 5. According to our results, grams size 2 produces better sentence selection. In general, the parameters that produced the best results are shown in Table 2.

Table 2. Parameters used in the tests with better results.

Feature	Parameter
Selection operator	Roulette
Crossover operator	Crosses with priority on common genes 100%
Mutation operator	Double inversion mutation 0.019%
Elitism	50 and 200 words 0.02%, 100 words: 0.03%
Number of generations	150
Number of individuals	<i>Number of sentences by 9</i>

In [5] was realized an analysis of slope in, concluding when the slope value is negative the first sentences are more important. Contrariwise, if the slope value is 0, all sentences have the same importance. Due to this reason, in our experimentation, we have used tests with slope values from -0.1 to -1 . To determine which slope value was best for each length, the best results are presented in Table 3.

As can be seen from the results obtained, when the summaries are created at a short length, the value of the slope that produced the best results is -0.1 . According to [5, 16], this means that all the sentences of the meta-document have the same importance. While the size of the summary increases, the sentences that are considered important are found close to the beginning of the text. From the results obtained for the length of 100 words, the value of the slope was -0.6 . While for summaries of 200 words, the value of the slope was -0.8 . It means that the most important content is in the first sentences.

Table 3. Results with several values of slope.

Values of slope	50 words		100 words		200 words	
	Rouge-1	Rouge-2	Rouge-1	Rouge-2	Rouge-1	Rouge-2
-0.1	28.023	6.861	32.762	7.185	39.243	9.608
-0.2	27.774	6.544	32.577	7.318	39.892	9.986
-0.3	26.853	6.117	33.100	7.473	39.939	9.957
-0.4	26.430	6.039	33.249	7.475	39.761	9.959
-0.5	26.931	5.888	33.459	7.638	39.088	9.988
-0.6	26.726	6.132	34.451	8.023	39.789	10.131
-0.7	27.033	5.584	32.937	7.391	40.039	10.087
-0.8	27.337	6.429	32.499	6.817	41.008	10.607
-0.9	26.974	5.632	32.765	7.298	40.370	10.521
-1.0	27.259	5.907	32.980	7.233	39.826	10.136

4.4 Description and Comparison of the State-of-the-Art Methods and Heuristics

To examine the performance of the proposed method was compared with state-of-the-art methods and heuristics. Supervised methods were not considered in the following analysis because the proposed method generates summaries from the information given in source documents, so it does not require external resources such as corpora, dictionaries, thesaurus, and lexicons. That is, it works in an unsupervised way.

CBA: In [22] was proposed a clustering-based method for MDTS. K-means were used in clustering. To define the sentences that should be selected for the final summary. Moreover, the sequence in which it will appear. The clustering was ranked via a cosine similarity measure.

NeATS: Lin and Hovy [23] proposed an Extractive MDTS system. The textual features such as term frequency, sentence position, stigma words, and a simplified version of Maximum Marginal Relevance were applied to choose filter content.

LexPageRank: In this method, the importance of sentences was computed based on the idea of centrality in a graph representation of sentences. In this, the connectivity matrix is based on cosine similarity [24].

GA-1: This method model MDTS like an optimization problem through GA[15].

Topline: The authors calculated the upper bounds in this work, which is possible to achieve by state-of-the-art methods [10, 25].

Baseline-First: It takes the first sentence from the document collection in chronological sequence until the target summary size is fulfilled [15].

Baseline-Random: This randomly selects sentences to incorporate them as an extractive summary until the length is required [10, 15].

Baseline-First Document: It includes the first 50, 100, and 200 words from the first document of a set of them until the target summary size is fulfilled [15].

Lead Baseline: This takes the first 50, 100, and 200 words from the last document in the set, where documents are supposed to be chronologically prepared [15].

We have compared the obtained results of the proposed method to other state-of-the-art methods and heuristics. In the comparison, the values Rouge-1 and Rouge-2 are exposed. Also, there is a comparison of the level of advance between the state-of-the-art methods and heuristics. To compute the performance, we use the formula (see Eq. 4), based on the assumption that the performance of the Topline heuristic is 100% and Baseline-random is 0% [25].

$$\%Advanced = \frac{(Rouge1_{Method} - Rouge1_{Baseline-Random}) * 100}{Rouge1_{Topline} - Rouge1_{Baseline-Random}} \quad (4)$$

Tables 4, 5, and 6 show this comparison using different summary lengths.

In the task where the summary length is 50 words (see Table 4), with the proposed method, the preceding results were improved by 12.7%, and the previous best result was the baseline-first document.

Table 4. Comparison of the state-of-the-art methods and heuristics, 50 words.

Method	Rouge-1	Rouge-2	Advanced (%)
Topline [25]	40.395	15.648	100.00%
Proposed	28.023	6.861	39.25%
Baseline-first document	25.435	4.301	26.55%
Baseline-first	25.194	4.596	25.36%
CBA [22]	22.679	2.859	13.02%
Lead Baseline	22.620	4.341	12.73%
NeATS [23]	22.594	2.963	12.60%
Baseline-random	20.027	1.929	00.00%

On the other hand, where the summary length is 100 words (see Table 5), the improvement is 6.08% with respect to what was considered the best result, which was Lex-PageRank method. As can be seen, in this length of summaries, there is a method whose performance, according to Eq. 4, is below the Baseline-random heuristic considered as the worst selection of sentences.

For the summary length is 200 words (see Table 6), the improvement was 4.01% more than the best method reported, which was GA-1. At this length, the heuristics have a better performance than in the 100 words task due to outperforming Baseline-Random, except Lead-Baseline, whose performance is even a negative value.

Table 5. Comparison of the state-of-the-art methods and heuristics, 100 words.

Method	Rouge-1	Rouge-2	Advanced (%)
Topline [25]	47.256	18.994	100.00%
Proposed	34.451	8.023	36.80%
LexPageRank [24]	33.220	5.760	30.72%
Baseline-first	31.716	6.962	23.30%
Baseline-first document	30.462	5.962	17.11%
NeATS [23]	28.195	4.037	05.92%
Lead Baseline	28.195	4.109	05.92%
Baseline-random	26.994	3.277	00.00%
CBA [22]	26.741	3.510	-01.24%

Table 6. Comparison the state-of-the-art methods and heuristics, 200 words.

Method	Rouge-1	Rouge-2	Advanced (%)
Topline [25]	53.630	22.703	100.00%
Proposed	41.008	10.607	35.51%
GA-1 [15]	40.224	10.306	31.50%
Baseline-first	39.280	9.339	26.68%
NeATS [23]	37.883	7.674	19.54%
Baseline-first document [15]	35.472	7.225	7.22%
CBA [22]	34.108	5.525	0.26%
Baseline-random [15]	34.057	5.240	0.00%
Lead Baseline [15]	34.009	6.195	-0.24%

5 Conclusions

In this paper, we formalized the summarization of a set of documents as a combinatorial optimization problem. In particular, GA was introduced to satisfy the extraction of the most relevant content from a collection of documents by using textual features, such as coverage and sentence position. Moreover, we improve the performance by incrementing the population size to explore an optimal solution better. Finally, we perform different experiments on the available benchmark dataset DUC01 in the English language for the lengths of 50, 100, and 200 words. The results show that the method is competitive with state-of-the-art previously reported results. Also, the summaries produced by the proposed method have achieved high evaluation scores compared with abstract gold standard summaries without needing external data.

References

1. Gao, S., Chen, X., Ren, Z., Zhao, D., Yan, R.: From Standard Summarization to New Tasks and Beyond: Summarization with Manifold Information (2020)
2. Roul, R.K., Mehrotra, S., Pungaliya, Y., Sahoo, J.K.: A new automatic multi-document text summarization using topic modeling. In: Fahrnerberger, G., Gopinathan, S., Parida, L. (eds.) ICDCIT 2019. LNCS, vol. 11319, pp. 212–221. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05366-6_17
3. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: a comprehensive survey (2021). <https://doi.org/10.1016/j.eswa.2020.113679>
4. García-Hernández, R.A., Ledeneva, Y.: Single extractive text summarization based on a genetic algorithm. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Rodríguez, J.S., di Baja, G.S. (eds.) MCPFR 2013. LNCS, vol. 7914, pp. 374–383. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38989-4_38
5. Mendoza, G.A.M., Ledeneva, Y., García-Hernández, R.A.: Determining the importance of sentence position for automatic text summarization. *J. Intell. Fuzzy Syst.* **39**, 2421–2431 (2020). <https://doi.org/10.3233/JIFS-179902>
6. Over, P., Dang, H.: DUC in context. *Inf. Process. Manag.* **43**, 1506–1520 (2007). <https://doi.org/10.1016/J.IPM.2007.01.019>
7. NIST (National Institute of Standards and Technology): TAC 2008 Summarization Track. <https://tac.nist.gov/2008/summarization/>. Accessed 20 July 2020
8. Fabbri, A.R., Li, I., She, T., Li, S., Radev, D.R.: Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model (2019)
9. Lins, R.D., et al.: The CNN-Corpus: A large textual corpus for single-document extractive summarization. In: Proceedings of the ACM Symposium on Document Engineering, DocEng 2019, pp. 1–10. Association for Computing Machinery, Inc, New York, New York, USA (2019). <https://doi.org/10.1145/3342558.3345388>
10. Matias, G., Ledeneva, Y., García, R.: Detección de ideas principales y composición de resúmenes en inglés, español, portugués y ruso. 60 años de investigación. Alfaomega Grupo Editor, S.A. de C.V (2020)
11. Ma, C., Zhang, W.E., Guo, M., Wang, H., Sheng, Q.Z.: Multi-document Summarization via Deep Learning Techniques: A Survey (2020). <https://doi.org/10.1145/nnnnnnn.nnnnnnn>
12. Hou, S.-L., et al.: A survey of text summarization approaches based on deep learning. *J. Comput. Sci. Technol.* **36**(3), 633–663 (2021). <https://doi.org/10.1007/s11390-020-0207-x>
13. Ledeneva, Y., García-Hernández, R.A.: Generación automática de resúmenes Retos, propuestas y experimentos. Universidad Autónoma del Estado de México (2017)
14. Vázquez, E., García-Hernández, R.A., Ledeneva, Y.: Sentence features relevance for extractive text summarization using genetic algorithms. *J. Intell. Fuzzy Syst.* **35**, 353–365 (2018). <https://doi.org/10.3233/JIFS-169594>
15. Neri-Mendoza, V., Ledeneva, Y., García-Hernández, R.A.: Unsupervised extractive multi-document text summarization using a genetic algorithm. *J. Intell. Fuzzy Syst.* **39**, 2397–2408 (2020). <https://doi.org/10.3233/JIFS-179900>
16. Neri Mendoza, V., Ledeneva, Y., García-Hernández, R.A.: Abstractive multi-document text summarization using a genetic algorithm. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Olvera-López, J.A., Salas, J. (eds.) MCPFR 2019. LNCS, vol. 11524, pp. 422–432. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21077-9_39
17. Sastry, K., Goldberg, D., Kendall, G.: Chapter 4 Genetic Algorithms. (2005)
18. Du, K.L., Swamy, M.N.S.: Search and optimization by metaheuristics: techniques and algorithms inspired by nature (2016). <https://doi.org/10.1007/978-3-319-41192-7>
19. Borges, J.L.: La doctrina de los ciclos (2013)

20. Rojas-Simón, J., Ledeneva, Y., García-Hernández, R.A.: Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm. *Expert Syst. Appl.* **167**, 113827 (2021). <https://doi.org/10.1016/J.ESWA.2020.113827>
21. Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries (2004)
22. Boros, E., Kantor, P.B., Neu, D.J.: A Clustering Based Approach to Creating Multi-Document Summaries (2001)
23. Lin, C.-Y., Hovy, E.: From single to multi-document summarization. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL 2002, p. 457 (2002). <https://doi.org/10.3115/1073083.1073160>
24. Wang, D., Zhu, S., Li, T., Gong, Y.: Multi-document summarization using sentence-based topic models. In: ACL and AFNLP, p. 297 (2010). <https://doi.org/10.3115/1667583.1667675>
25. Rojas Simón, J., Ledeneva, Y., García Hernández, R.A.: Calculating the Upper Bounds for Multi-Document Summarization using Genetic Algorithms. *Comput. y Sist.* **22** (2018) <https://doi.org/10.13053/cys-22-1-2903>