

# Generic and Update Multi-Document Text Summarization based on Genetic Algorithm

Verónica Neri-Mendoza, Yulia Ledeneva,  
René Arnulfo García-Hernández, Ángel Hernández-Castañeda

Universidad Autónoma del Estado de México,  
Toluca de Lerdo,  
Mexico

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

**Abstract.** In this paper, we addressed the generic and update text summarization tasks of a set of documents as a combinatorial optimization problem through a genetic algorithm and unsupervised textual features. Particularly under the news domain, input documents are a set of articles of varying sizes covering the same event. The main advantage of the proposed method is that it is language-independent. The experimental results demonstrated that the method performs well for both kinds of summarization. Moreover, we calculated the heuristics for update text summarization like a benchmark to compare state-of-the-art methods.

**Keywords.** Generic, update, multi-document, text summarization, genetic algorithm.

## 1 Introduction

The Internet has been an essential tool in society due to the development of new channels to spread information of public interest. The news change constantly. Everyone with internet access can find out about current facts.

However, the most essential content could be involved in many documents, making it difficult to recover quickly [12, 13].

For this reason, a domain that has been the object of study in state-of-the-art are news. The different news sources that report on a particular event contain common information that construct the main facts.

Thus, Document Text Summarization (DTS) from multiple news articles is a valuable field of study since the volume of information is overgrowing [13].

The DTS is the research area aiming at creating a brief version (summary) condensed from the original document [26, 11]. This document transmits the main idea in a few words.

In this context, Multi-Document Text Summarization (MDTS) is essential to satisfy the information needs of various users.

In the literature, multiple datasets based on the news domain have been developed, such as Multi-News [13] and CNN [19], among others, to evaluate the effectiveness of state-of-the-art methods.

Furthermore, workshops have been developed in this area, like DUC (Document Understand Conference) [25] and TAC (Text Analysis Conference), which have provided datasets and benchmarks [6].

In this work, we focused on two main situations. First, generic MDTS using DUC01 which consists of when the user wants to know about a fact finished in the past to obtain a summary with all related information.

The main challenge of generic multi-document text summarization involves the selection of sentences from source documents evaluating the relevance of sentences, and ordering them to select the most important sentences [25].

In addition to generic MDTS, we have focused on the update MDTS using the TAC08 dataset.

The objective of this task is based on the idea that the user needs to be informed while the news is happening, getting two summaries.

The first one is the initial (summary A) which contains the introduction about a fact. The second one is the update (summary B), assuming that the user knows the topic and is provided an update of facts.

To produce an update summary, the method should select which information is novel in the documents, avoiding redundancy. This selection is crucial to generate summaries with a high update value [6].

Optimization-based approaches have been gaining importance because of the excellent performance obtained due to these being effective in getting optimal solutions for huge and varied spaces [11, 15].

These approaches generally are helpful to recognize appropriate sentences to construct summaries in the DTS context. In the literature of DTS, there are three approaches to generate text summaries: extractive, abstractive, and hybrid [14].

**Extractive Text Summarization.** Proposed systems based on this approach create summaries by assigning weights to sentences according to linguistic and statistical features, then those sentences with the best weights are selected.

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 [11, 21, 12].

**Abstractive Text Summarization.** This approach allows the proposed methods to create summaries using new corpus words and sentences. The process of abstractive summarization is like the human tends to generate summaries.

However, it requires sophisticated natural language understanding and generation techniques, such as paraphrasing and sentence fusion [21, 12].

**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 one 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 [11, 12].

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 [14, 11, 16].

We formulated the 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 that 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 generic summaries of 50, 100, and 200 words on the DUC01 dataset. On the other hand, we conducted experiments on the TAC08 dataset to create updated summaries of 100 words.

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

## 2 Related Work

The history of summarization has concentrated mainly on the production of generic summaries.

Over its first six years, DUC has examined MDTS of news articles through generic summarization, which has been the initial driver of research, and has formed the bulk of research up until recently.

A good generic summary always shows the information about every aspect mentioned in the collection of documents [25].

On the other hand, the update summarization task was first introduced in TAC08, which targets summarizing news documents in a continuously growing text stream, such as news events.

When a new document arrives in a sequence of events, the summary needs to be updated, considering previous information [6].

In the literature, the DTS is usually tackled through many techniques, such as supervised-based learning methods, addressing the summarization task into a supervised classification problem.

Generally, these methods learn by training to classify sentences, indicating whether a sentence is included in the summary.

In addition, state-of-the-art approaches usually use word embeddings to represent the contextual meaning of sentences. Nevertheless, the proposed methods require a manually tagged [13, 11].

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 [11, 20].

Four steps have been identified to generate a summary: term selection, term weighting, sentence weighting, and sentence selection [13, 11]. For the last step, various textual features have been developed [10]. Some of them are:

**Similarity with the title:** This feature assigns the most important relevance to the sentences that include many words in the title.

**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.

**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.

**Redundancy reduction:** Redundant or duplicate information in the generated summary is expected to be minimized.

**Sentence position:** The idea is that the first sentences indicate a relevant sentence.

**Coverage:** This feature is based on the idea that information provided in the original documents should be included in the generated summary.

Optimization-based approaches have been gaining importance because of the excellent performance obtained in the state-of-the-art [3].

These approaches represent the summarization problem, in an optimization problem as proposed in [24], which addresses the extractive summarization as a binary optimization problem using a modified quantum-inspired genetic algorithm.

In this work, the objective function is a linear combination of textual features like coverage, relevance, and redundancy.

Another work based on GA is proposed in [22]; which seeks to determine the optimal weights of text features, including sentence location, title similarity, sentence length, and positive and negative words.

Despite its competitive performance in comparison to other methods, both the proposed method and textual features have not been tested in more representative collections of documents.

### 3 Proposed Method

In MDTS, the search space is more extensive than the Single DTS, making selecting the most important sentences more challenging.

Therefore, the documents of each collection are considered a set of sentences, and the aim is to choose an optimal subset from sentences under a length constraint.

Previous works [23] 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. Population size is an essential factor that usually affects the GA performance [15].

Small population sizes might generally lead to premature convergence and yield substandard solutions [9].

### 3.1 Pre-processing

In this step, the documents of each collection were ordered chronologically, 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 [20].

### 3.2 Text Modeling

After preprocessing the text, it is necessary to perform the text modeling. This stage aims to predict the probability of natural word sequences. The simplest and most successful form of text modeling is the n-gram, which is a text representation model that constructs contiguous subsequences of consecutive words from a given text [20].

### 3.3 Weighting and Selection of Sentences

Sentence weighting and selection of sentences usually worked together, 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 is 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 [11, 23].

**Generation of population:** The initial population is randomly generated. On the other hand, the population of the next generations is generated from the selection, crossover, and mutation stages.

The search process concludes when a termination criterion is met. Otherwise, a new generation will be produced, and the search process will continue [20].

**Size of the initial population:** The size of the population is determined according to the number of sentences from the meta-document [15, 20].

**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 [20].

**Crossover Operator:** In the crossover, it is performed an exchange of genes of each pair of selected parents. Nevertheless, of the two randomly selected pairs of parents, only genes are chosen randomly with a gene that will belong to the offspring.

The genes with value 1 in both parents are more likely to be chosen, in order to meet the condition of the minimum number of words. If each gene is selected, it will be part of the offspring, then the number of words is counted.

**Mutation Operator:** We use the flipping operator, which consist of changing the value of each gene, inverting from 1 to 0 or vice versa. However, the mutation is performed by 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 [20].

**Fitness function:** It is calculated by employing the concept of the slope of the line. This 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  $t$  the slope to be discovered. The formula to calculate the importance of the sentence position is shown in Equation 1:

$$\text{Sent. Imp.} = \frac{\sum_{j=1}^n 1^{t(i-x)+x}}{|c_j|} x = 1 + \frac{(n-1)}{2}, \quad (1)$$

where  $k$  is the number of selected sentences. On the other hand, the content coverage was calculated using the summation of the frequencies of the n-grams that the summary has included.

Thus (Precision-Recall) was calculated via the sum of the frequencies of the n-grams considered in the original text divided by the sum of the frequencies of the different n-grams of summary (see Equation 2):

$$\text{Precision} - \text{Recall} = \frac{\sum \text{Original text frequency}}{\sum \text{Frequency summary}}, \quad (2)$$

Finally, to obtain the value of the fitness function, the following formula was applied, which is multiplied by 1000 [20]:

$$\text{FA} = \text{Precision} - \text{Recall} \times \text{Sentence Importance}. \quad (3)$$

**Stop condition:** For this step, we have used the number of generations as a stop condition. Below, it is shown the pseudocode of the proposed GA:

---

#### Algorithm 1 GA Summarization

$\triangleright (G, P_m, P_c, E, MD)$

---

```

1:  $P_0 \leftarrow \text{Random}(S_p)$ 
2:  $\triangleright (S_p) \leftarrow \text{Number of Sentences} \times 09$ 
3:  $\triangleright (S_p) \leftarrow \text{Number of Sentences} \times 12$ 
4: for  $i = 1$  to  $G$  do
5:    $P_i \leftarrow \text{FA}(\text{Equation 3}(MD))$ 
6:    $P_i \leftarrow \text{Roulette}(\text{FA}(P_i), P_i)$ 
7:    $P_i \leftarrow \text{X Uniform}(P_i, P_c)$ 
8:    $P_i \leftarrow \text{FlippingM}(P_i, P_m)$ 
9:    $P_E \leftarrow \text{Elitism}(P_i, E)$ 
10:   $P_i \leftarrow (P_E \cup P_i)$ 
11: end for

```

---

In the GA, it is required the number of Generations ( $G$ ), mutation probability ( $P_m$ ), crossover probability ( $P_c$ ), Elitism ( $E$ ), and the Meta-Document representation ( $MD$ ).

First, the initial population is generated randomly. The population size is determined according to the number of sentences from the Meta-document.

In the case of the generic dataset, the number of individuals is multiplied by 9. On the other hand, for the update dataset, the number of individuals is multiplied by 12.

After this, the population is evaluated by the fitness function using  $MD$  as input.

After this, the roulette selection operator is applied to the population. The next step is to use the uniform crossover operator (X Uniform) and the  $P_c$  is required.

Afterward, it is carried out the mutation by flipping operator, in which is required the population and mutation probability ( $P_m$ ). Moreover, some individuals are selected by elitism and the population is updated till  $i$  reaches  $G$ .

## 4 Experimental and Results

### 4.1 Dataset

To empirically evaluate the results of the proposed method in generic MDTs, we use the DUC01 dataset. This 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 50, 100, and 200 words. In addition, this dataset includes two gold standard summaries for each collection [25, 17].

We chose this dataset because the gold standards summaries provided in it were written in an abstractive way.

This allowed us to measure how competitive the proposed extractive unsupervised method can be over summaries made by paraphrases, words, and sentences that do not belong to source documents [6, 17].

Moreover, to evaluate the update MDTs task, we employed the TAC08 dataset, which comprises 48 topics, each having 20 documents divided into two sets, A and B.

Set A chronologically precedes the documents in set B. The length of created summaries is 100 words, and each one is compared with four gold standard summaries provided in the dataset [7].

We evaluate in generic and update MDTs tasks to test the performance of the proposed method in two different stages because the users have different needs to satisfy their knowledge about a specific fact.

**Table 1.** Parameters used in the tests with better results for generic and update MDTs

| Feature                      | Generic MDTs |           |            | Update MDTs |           |
|------------------------------|--------------|-----------|------------|-------------|-----------|
|                              | 50 words     | 100 words | 2000 words | Summary A   | Summary B |
| Selection operator           | Roulette     | Roulette  | Roulette   | Roulette    | Roulette  |
| Crosses                      | 100%         | 100%      | 100%       | 99%         | 99%       |
| Double inversion mutation    | 0.019%       | 0.019%    | 0.019%     | 0.010%      | 0.020%    |
| Elitism                      | 0.02%        | 0.03      | 0.02%      | 0.02%       | 0.00%     |
| Number of generations        | 150          | 150       | 150        | 400         | 400       |
| Size of population ( $S_p$ ) | $NS*9$       | $NS*9$    | $NS*9$     | $NS*12$     | $NS*12$   |

## 4.2 Evaluation Measures

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an evaluation system that uses several measures to automatically establish the quality of a summary created by a proposed method by contrasting it to other ideal summaries created by humans (gold standard summaries) [18].

For instance, ROUGE-N count the number of overlapping n-grams, between the computer-generated summary and a set of ideal summaries created by humans [5, 17].

Especially, we have employed the evaluation under unigrams (ROUGE-1) and bigrams (ROUGE-2 and ROUGE-SU4) to evaluate summaries. The results obtained were compared through F-measure of ROUGE-1 y ROUGE-2 for generic MDTs.

While for the TAC08 dataset, we evaluated summaries under two versions of ROUGE: ROUGE-2 and ROUGE-SU4; due to these measures have been considered as standard evaluations in TAC08 [6].

## 4.3 Parameter Selection

We perform tests with different parameters such as roulette selection operator, uniform crossover, flipping mutation, with different crossover and mutation probabilities, respectively.

Also, we conducted experiments by 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. In general, the parameters that produced the best

results are shown in Table 1 for both generic and update MDTs.

According to the obtained results, we observe a better GA exploration for the different generic summaries lengths (50, 100, and 200 words) by incrementing the population size using the product between the number of sentences from the meta-document and 9 over 150 generations.

Moreover, these parameters favored the selection process, obtaining better fitness values by the roulette operator. For update summaries, we obtained better results, multiplying the number of sentences in the meta-document by 12 over 400 generations.

Moreover, we tested n-grams of sizes from 1 to 5. According to our results, n-gram lengths equals 2 produces better sentence selection in both generic and update summaries. In [20], it was performed an analysis of slope for generating summaries, concluding when the slope value is negative, the first sentences show the most important content.

Contrariwise, if the slope value is 0, all sentences have the same importance. Due to this reason, in our experimentation, we have employed slope values from -0.1 to -1. To determine which slope value was best for each summary length, we compared slope values in table 2.

As we can observe, when the summaries are created with shorter lengths, the slope value that produced the best results is -0.1. According to [20, 23], this means that the last sentences of the meta-document have relevant content.

While the size of the summary increases, the sentences that are considered important are found

**Table 2.** Results with several values of slope for generic and update MDTs

| slope value | Generic MDTs  |              |               |              |               |               | Update MDTs   |               |              |               |
|-------------|---------------|--------------|---------------|--------------|---------------|---------------|---------------|---------------|--------------|---------------|
|             | 50 words      |              | 100 words     |              | 200 words     |               | Summary A     |               | Summary B    |               |
|             | R-1           | R-2          | R-1           | R-2          | R-1           | R-2           | R-2           | R-SU4         | R-2          | R-SU4         |
| -0.1        | <b>28.023</b> | <b>6.861</b> | 32.762        | 7.185        | 39.243        | 9.608         | 10.894        | 13.247        | <b>9.709</b> | <b>12.736</b> |
| -0.2        | 27.774        | 6.544        | 32.577        | 7.318        | 39.892        | 9.986         | 10.877        | 13.240        | 9.503        | 12.591        |
| -0.3        | 26.853        | 6.117        | 33.100        | 7.473        | 39.939        | 9.957         | <b>10.909</b> | <b>13.249</b> | 8.953        | 12.137        |
| -0.4        | 26.430        | 6.039        | 33.249        | 7.475        | 39.761        | 9.959         | 10.485        | 12.938        | 8.429        | 11.446        |
| -0.5        | 26.931        | 5.888        | 33.459        | 7.638        | 39.088        | 9.988         | 9.074         | 12.123        | 8.465        | 11.655        |
| -0.6        | 26.726        | 6.132        | <b>34.451</b> | <b>8.023</b> | 39.789        | 10.131        | 8.832         | 12.154        | 8.589        | 11.736        |
| -0.7        | 27.033        | 5.584        | 32.937        | 7.391        | 40.039        | 10.087        | 8.895         | 12.009        | 8.547        | 11.738        |
| -0.8        | 27.337        | 6.429        | 32.499        | 6.817        | <b>41.008</b> | <b>10.607</b> | 9.211         | 12.253        | 8.230        | 11.403        |
| -0.9        | 26.974        | 5.632        | 32.765        | 7.298        | 40.370        | 10.521        | 8.871         | 11.786        | 8.045        | 11.326        |
| -1.0        | 27.259        | 5.907        | 32.980        | 7.233        | 39.826        | 10.136        | 8.904         | 12.074        | 7.740        | 11.052        |

close to the beginning of the text. For summaries of 100 words, the slope value was -0.6.

While for summaries of 200 words, the slope value was -0.8. It means that the most important content is in the first sentences.

In the table 2, we observe that the slope value used to generate summaries of 100 words for update MDTs task is different from generic MDTs, even the summaries have the same length.

According to these results, in update MDTs task, the context of the news is close to ending of documents (summary A).

On the other hand, in update summaries (summary B), the essential information is in the last part of each source document. Therefore, the novelty is recognized at the end of documents.

#### 4.4 Analysis of the Results

To examine the performance of the proposed method, it was compared to 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, lexicons, or a previous training.

That is, it works in an unsupervised way. We have compared the obtained results of the proposed method to other state-of-the-art methods and heuristics.

The values ROUGE-1 and ROUGE-2 for generic MDTs and ROUGE-2 and ROUGE-SU4 for update MDTs are exposed.

Also, it is provided a comparison of the level of advance between the state-of-the-art methods and heuristics.

To compute the performance, we use the formula shown in Equation 4, based on the assumption that the performance of the Topline heuristic is 100% and Baseline-random is 0% [27]:

$$\% \text{ Advanced} = \frac{(R1_{\text{Method}} - R1_{\text{BaselineRandom}}) \times 100}{R1_{\text{Topline}} - R1_{\text{BaselineRandom}}} \quad (4)$$

##### 4.4.1 State-of-the-Art Heuristics

**Topline:** The authors calculated the upper bounds for MDTs via GA, which is possible to achieve by state-of-the-art methods [27].

**Baseline-first:** It takes the first sentences from the document collection in chronological sequence until the target summary size is fulfilled [23].

**Table 3.** Comparison of the state-of-the-art methods and heuristics, generic MDTs

| Method          | 50 words |        |         | 100 words |        |         | 200 words |        |         |
|-----------------|----------|--------|---------|-----------|--------|---------|-----------|--------|---------|
|                 | R-1      | R-2    | Adv(%)  | R-1       | R-2    | Adv(%)  | R-1       | R-2    | Adv(%)  |
| Topline         | 40.395   | 15.648 | 100.00% | 47.256    | 18.994 | 100.00% | 53.630    | 22.703 | 100.00% |
| Proposed        | 28.023   | 6.861  | 39.25%  | 34.451    | 8.023  | 36.80%  | 41.008    | 10.607 | 35.51%  |
| BFD             | 25.435   | 4.301  | 26.55%  | 30.462    | 5.962  | 17.11 % | 35.472    | 7.225  | 7.22%   |
| Baseline-first  | 25.194   | 4.596  | 25.36%  | 31.716    | 6.962  | 23.30%  | 39.280    | 9.339  | 26.68%  |
| CBA             | 22.679   | 2.859  | 13.02%  | 26.741    | 3.510  | -01.24% | 34.108    | 5.525  | 0.26%   |
| Lead Baseline   | 22.620   | 4.341  | 12.73%  | 28.195    | 4.109  | 05.92%  | 34.009    | 6.195  | -0.24%  |
| NeATS           | 22.594   | 2.963  | 12.60%  | 28.195    | 4.037  | 05.92%  | 37.883    | 7.674  | 19.54%  |
| Baseline-random | 20.027   | 1.929  | 00.00%  | 26.994    | 3.277  | 00.00%  | 34.057    | 5.240  | 0.00%   |
| LexPageRank     | -        | -      | -       | 33.220    | 5.760  | 30.72%  | -         | -      | -       |

**Baseline-random:** This heuristic randomly selects sentences to incorporate them as an extractive summary until the length required [23].

**Baseline-first document (BFD):** It includes the first 50, 100, and 200 words from the first document from a set of them until the target summary size is fulfilled [23].

**Lead Baseline:** This heuristic takes the first 50, 100, and 200 words from the last document to construct extractive summaries. The documents are supposed to be chronologically sorted [23].

#### 4.4.2 State-of-the-Art Generic MDTs Methods

**CBA:** In [4], it was proposed a clustering-based method for MDTs, using the K-means algorithm to define the sentences that should be selected for the final summary. The clustering was performed via a cosine similarity measure.

**NeATS:** Lin and Hovy proposed in [5] an extractive MDTs system whose functionality is based on textual features such as term frequency, sentence position, stigma words, and a simplified version of Maximum Marginal Relevance to choose and filter relevant sentences.

**LexPageRank:** In this method [8], the importance of sentences was computed based on the idea of centrality in a graph representation of sentences.

In the task where the summary length is 50 words (see Table 3), with the proposed method, the preceding results were improved by 12.7%, and the previous best result was the baseline-first document.

On the other hand, where the summary length is 100 words (see Table 3), the improvement is 6.08% with respect to what was considered the best result, which was LexPageRank method.

As can be seen, in this length of summaries, there is a method whose performance, according to equation 4, is below the Baseline-random heuristic considered as the worst selection of sentences.

For the summary length is 200 words (see Table 3), the improvement was 8.83% more than the best method reported, which was baseline-first heuristic.

At this length, the proposed method have a better performance due to it outperforms Baseline-Random and Lead-Baseline, whose performance is even a negative value.

#### 4.4.3 State-of-the-Art Update MDTs Methods

The TAC08 summarization workshop had 33 participant systems worldwide; they submitted 71 different results. For practical purposes, we only have considered the systems with the best



**Table 4.** Comparison of the state-of-the-art methods and heuristics, Update MDTS

| Method          | Summary A |        |          | Summary B |        |         |
|-----------------|-----------|--------|----------|-----------|--------|---------|
|                 | R-2       | R-SU4  | Adv      | R-2       | R-SU4  | Adv     |
| Topline         | 15.607    | 19.780 | 100.00%  | 16.119    | 19.839 | 100.00% |
| Proposed        | 10.909    | 13.249 | 58.622%  | 9.709     | 12.736 | 46.707% |
| ICSI            | 10.900    | 14.100 | 58.543%  | 9.400     | 12.700 | 44.138% |
| Abawakid        | 7.900     | 11.500 | 32.120%  | 8.100     | 11.900 | 33.330% |
| BFD             | 6.512     | 9.858  | 19.896 % | 5.502     | 9.378  | 11.730% |
| Baseline-first  | 5.851     | 9.035  | 14.074 % | 6.359     | 9.578  | 18.856% |
| Lead Baseline   | 5.800     | 9.300  | 13.625%  | 6.000     | 9.400  | 15.871% |
| LIPN            | 4.400     | 8.400  | 1.294%   | 3.400     | 7.000  | -5.744% |
| Baseline-random | 4.253     | 8.404  | 0.000%   | 4.091     | 8.359  | 0.000%  |

(ICSI), the median (Abawakid), and the lowest (LIPN) performance.

Moreover, the organizers (NIST) created a baseline automatic summarizer, which selects the first sentences of the most recent document, such that the generated summary do not exceed 100 words (Lead Baseline).

To have a broader comparison, we have calculated the Topline through a GA using the parameters provided in [27], in order to know the possible best result of sentence combinations. In addition, we have computed heuristics, such as the baseline-first, baseline first document, and baseline-random.

**ICSI:** This system was proposed in [7] is based on a general framework that casts summarization as a global optimization problem with an integer linear programming solution.

**Abawakid:** This system used a scoring function to identify the most relevant sentences from a set of documents. In addition, this system uses sentence features such as the position of sentences, sentences location, sentence-sentence similarity [1].

**LIPN:** In [2], a fast global K-means was employed to compute sentence similarity to detect novelty between the A and B summaries.

The authors considered that sentences are not conveying novelty if they are closer to sentences belonging to the first document set than sentences belonging to the second document set.

The comparison among participants systems at the TAC08 workshop, the heuristics, and the proposed method is shown in the Table 4.

The proposed method shows improvements over ICSI, which has been the best method in the TAC08 workshop. According to the ROUGE-2 results between the proposed method and ICSI for summary A, the enhancement is 0.079%.

However, ICSI shows a better performance than the proposed method under the ROUGE-SU4 measure. Besides, the proposed method shows better results than ICSI for updated summaries (summary B), showing 2.569% of improvement under ROUGE-2.

In general, the results obtained are close to ICSI, but, the proposed method had a better performance. The heuristics that we calculated allow us to support this asseveration.

In addition, it is worth mentioning that even the lowest results of the proposed method, shown in Table 2 overcomes the heuristics (except the Topline), Abawakid, and LIPN systems.

## 5 Conclusions

In this paper, we have addressed the MDTS task as a combinatorial optimization problem based on GA. We used coverage and sentence position as features, which allowed us to retrieve important aspects of content in a collection.

We tested the performance of the proposed method in generic and update summarization on DUC01 and TAC08. Furthermore, we experimented with several parameters and used the complete collections.

On DUC01, in 3 different lengths, the summaries produced by the proposed method have achieved high evaluation scores compared with abstract gold standard summaries without needing external data.

For TAC08, the proposed method shows results close to the best method that participated in the workshop for summaries of type A. On the other hand, the proposed method shows improvements over state-of-the-art methods in generating updated summaries (B), as shown in Table 4. Therefore, it manages the content of documents to recognize novelty.

Moreover, we demonstrated that the proposed GA method is flexible because making changes in some parameters keeps high performance in generic and update MDTs.

## References

1. **Abdullah, B., Oussalah, M. (2008).** A semantic summarization system: University of birmingham at tac 2008. TAC, pp. 1506–1520.
2. **Aurélien, B., Michel, G., Poibeau, T. (2008).** Description of the lipn system at tac 2008: Summarizing information and opinions. TAC 2008, pp. 282–291.
3. **Bidoki, M., Fakhrahmad, M., Moosavi, M. R. (2020).** Text summarization as a multi-objective optimization task: Applying harmony search to extractive multi-document summarization. The Computer Journal, Vol. 65, No. 5, pp. 1053–1072. DOI: 10.1093/comjnl/bxaa139.
4. **Boros, E., Kantor, P., Neu, D. (2002).** A clustering based approach to creating multi-document summaries. Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, pp. 74–81.
5. **Chin-Yew, L. (2004).** Rouge: A package for automatic evaluation of summaries. Text summarization branches out, pp. 74–81.
6. **Dang, H., Owczarzak, K. (2009).** Overview of the tac 2008 update summarization task.
7. **Daniel, G., Benoit, F., Hakkani-Tür, D. (2008).** The icsi summarization system at tac 2008. Tac, pp. 1506–1520.
8. **Dingding, W., Shenghuo, Z., Tao, L., Gong, Y. (2010).** Multi-document summarization using sentence-based topic models. ACL and AFNLP, pp. 297. DOI: 10.3115/1667583.1667675.
9. **Du Ke, L., Swamy, M. (2016).** Search and optimization by metaheuristics. DOI: 10.1007/978-3-319-41192-7.
10. **Eder, V., García-Hernández, R. A., Yulia, L. (2018).** Sentence features relevance for extractive text summarization using genetic algorithms. Journal of Intelligent and Fuzzy Systems, Vol. 35, No. 1, pp. 353–365. DOI: 10.3233/JIFS-169594.
11. **El-Kassas, W. S., Salama, C. R., Rafea, A. A., Mohamed, H. K. (2021).** Automatic text summarization: A comprehensive survey. Expert Systems with Applications, Vol. 165. DOI: 10.1016/j.eswa.2020.113679.
12. **Elsaid, A., Mohammed, A., Ibrahim, L. F., Sakre, M. M. (2022).** A comprehensive review of arabic text summarization. IEEE Access, Vol. 10, pp. 38012–38030. DOI: 10.1109/ACCESS.2022.3163292.
13. **Fabbri, A. R., Li, I., She, T., Li, S., Radev, D. R. (2019).** Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. Arxiv. DOI: 10.48550/arXiv.1906.01749.
14. **Gao, S., Chen, X., Ren, Z., Zhao, D., Yan, R. (2020).** From standard summarization to new tasks and beyond: Summarization with manifold information. Arxiv. DOI: 10.48550/ARXIV.2005.04684.
15. **García-Hernández, R. A., Ledenev, Y. (2013).** Single extractive text summarization based on a genetic algorithm. LNCS, Vol. 7914, pp. 374–383.
16. **Hou, S. L., Huang, X. K., Fei, C. Q., Zhang, S. H., Li, Y. Y., Sun, Q. L., Wang, C. Q. (2021).** A survey of text summarization approaches based on deep learning. Journal of Computer Science and Technology, Vol. 36, No. 3, pp. 633–663. DOI: 10.1007/s11390-020-0207-x.
17. **Jonathan, R. S., Yulia, L., García Hernández, R. A. (2022).** Background of the ETS, chapter 2. pp. 11–71. DOI: 10.1007/978-3-031-07214-7.2.
18. **Jonathan, R. S., Yulia, L., García Hernández, R. A. (2021).** Evaluation of text summaries without human references based on the linear optimization of content metrics using a genetic algorithm. Expert

- Systems with Applications, Vol. 167, pp. 113827. DOI: 10.1016/j.eswa.2020.113827.
19. **Lins, R. D., Oliveira, H., Cabral, L., Batista, J., Tenorio, B., Ferreira, R., Lima, R., de França Pereira e Silva, G., Simske, S. J. (2019).** The CNN-corpus: A large textual corpus for single-document extractive summarization. Proceedings of the ACM Symposium on Document Engineering, pp. 1–10. DOI: 10.1145/3342558.3345388.
  20. **Matias Mendoza, G. A., Ledeneva, Y., García-Hernández, R. A. (2020).** Determining the importance of sentence position for automatic text summarization. Journal of Intelligent and Fuzzy Systems, Vol. 39, No. 2, pp. 2421–2431. DOI: 10.3233/JIFS-179902.
  21. **Matias Mendoza, G. A., Ledeneva, Y., García Hernández, R. A. (2020).** Deteccion de ideas principales y composicion de resúmenes en inglés, español, portugués y ruso. 60 años de investigación.
  22. **Meena, Y. K., Gopalani, D. (2015).** Evolutionary algorithms for extractive automatic text summarization. Procedia Computer Science, Vol. 48, pp. 244–249. DOI: 10.1016/j.procs.2015.04.177.
  23. **Mendoza, V. N., Ledeneva, Y., García-Hernández, R. A. (2019).** Abstractive multi-document text summarization using a genetic algorithm. In Mexican Conference on Pattern Recognition. pp. 422–432. DOI: 10.1007/978-3-030-21077-9\_39.
  24. **Mojrián, M., Mirroshandel, S. A. (2021).** A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: Mtsqiga. Expert Systems with Applications, Vol. 171, pp. 114555. DOI: 10.1016/j.eswa.2020.114555.
  25. **Paul, O., Hoa, D., Donna, H. (2007).** Duc in context. Information Processing and Management, Vol. 43, No. 6, pp. 1506–1520.
  26. **Rajendra, K. R. (2021).** Topic modeling combined with classification technique for extractive multi-document text summarization. Soft Computing, Vol. 25, pp. 1113–1127. DOI: 10.1007/s00500-020-05207-w.
  27. **Rojas Simón, J., Yulia, L., García Hernández, R. A. (2018).** Calculating the upper bounds for multi-document summarization using genetic algorithms. Computación y Sistemas, Vol. 22, No. 1.

*Article received on 21/08/2022; accepted on 01/12/2022.  
Corresponding author is Yulia Ledeneva.*