A Generic Extractive Multi-document Text Summarization Method Using Memetic Algorithm and Combinatorial Optimization

Alysson Guimarães[1], **Methanias Colaço Junior**[1,2]

[1] Postgraduate Program in Computer Science (PROCC)
Federal University of Sergipe (UFS)
São Cristóvão – SE – Brazil

[2]Health Technological Innovation Laboratory (LAIS)
Onofre Lopes University Hospital
Federal University of Rio Grande do Norte (UFRN)
Natal – RN – Brazil

alyssonalk@gmail.com, mjrse@hotmail.com

***Abstract. Research Context**:Automatic text summarization remains a subject of considerable relevance across multiple domains. In particular, extractive multi-document generic summarization has garnered increased attention due to its capacity to mitigate information overload in a wide range of applications. **Scientific and/or Practical Problem**: The volume of unstructured text data produced on the internet has grown exponentially in recent years, driven by advances in information and communication technologies (ICTs). This massive generation of data makes it difficult for users to find relevant information. **Proposed Solution and/or Analysis**: This study introduces, implements, and applies the memetic algorithm known as Holistic Text Summarization with the Shuffled Frog-Leaping Algorithm (HSSFLA) to address the generic extractive multi-document multi-language text summarization problem using combinatorial optimization techniques. **Related IS Theory**: This research integrates swarm intelligence, memetic algorithms and combinatorial optimization. **Research Method**: An in vitro experiment was conducted to quantitatively compare the summary quality between the proposed method and similar methods in the literature. **Summary of Results**: Experiments were carried out on the DUC2001/2002 benchmark datasets, and performance was evaluated using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric. The results demonstrate that the proposed approach yielded an average improvement of 25.12% in ROUGE-1 and 34.91% in ROUGE-2 on the DUC 2001 dataset. On the DUC2002 dataset, the method achieved average gains of 35.42% in ROUGE-1 and 36.08% in ROUGE-2. **Contributions and Impact to IS area**: HSSFLA, a memetic algorithm based on swarm intelligence, was developed to solve this problem for the first time. It creates holistic summaries, in which it evaluates the quality of the summary as a whole, rather than focusing exhaustively on finding the best individual sentences. HSSFLA outperforms the results of the scientific literature in DUC2001 and DUC2002.*

1. Introduction

The volume of unstructured data, such as text, produced on the internet has grown exponentially in recent years, driven by advances in information and communication technologies (ICTs) and the rise of social networks. This massive generation of data over time makes it increasingly difficult for users to find relevant information on specific topics. However, by leveraging text mining tools such as automatic text summarization (ATS), it is possible to extract specific and relevant information from a dataset [Sanchez-Gomez et al. 2022]. Automatic summarization addresses this need by significantly reducing the volume of information while preserving the most relevant content of interest to the user.

In the scientific literature, automatic summaries can be generated in different ways. Based on the method used, summaries can be either abstractive or extractive. An abstractive summary generates new content that does not exist in the original document, creating novel words and sentences. In contrast, an extractive summary selects a subset of sentences from the original text to form the summary [Jorge et al. 2025]. Furthermore, summaries can be classified as generic or query-oriented. Generic summaries do not require any user input [Alguliev et al. 2012, Sanchez-Gomez et al. 2018, Abbasi-ghalehtaki et al. 2016], whereas query-oriented summaries require some form of user-provided information, typically a query or topic of interest in sentence form [Alguliev et al. 2012, Huang et al. 2010, Sanchez-Gomez et al. 2024]. Additionally, summarization methods can be categorized as single-document or multi-document. Single-document methods condense the information from a single text into a concise summary, while multi-document methods extract key information from a set of documents [Saini et al. 2019, Sanchez-Gomez et al. 2018, Mendoza et al. 2014]. Summarization approaches can also be classified as supervised or unsupervised [Alguliyev et al. 2015]. In supervised approaches, text summarization is treated as a classification problem, where a model identifies sentences to be included in the summary. However, these models require labeled training samples. Unsupervised methods, on the other hand, employ clustering algorithms to score sentences based on predefined features.

In recent years, there has been a shift in the focus of research in the field of automatic text summarization from extractive to abstractive approaches. This shift was driven by the development of the Transformer architecture, capable of generating natural language, enabling the emergence of Large Language Models (LLMs). Despite advances in the field, most research focuses on the English language, and few studies have been dedicated to Portuguese, especially for abstractive summarization. This gap limits the applicability of automatic summarization systems designed specifically for Portuguese, which lacks models and databases to support such research [Sarmento and de Oliveira 2024, Jorge et al. 2025]. Although abstractive approaches potentially generate summaries closer to human-like results, their development requires intensive computational resources. Nevertheless, extractive summarization approaches continue to be widely explored [Gomes and Oliveira 2019].

This paper focuses on the problem of generic extractive multi-document multi-language text summarization. In this study, the Holistic Text Summarization with Shuffle Frog-Leaping Algorithm (HSSLF) is developed, implemented, and applied to solve the problem of generic extractive multi-document multi-language text summarization.

In vitro experiments were conducted using the DUC 2001 and DUC 2002 datasets [DUC 2024]. The results were evaluated using the standard measure for assessing automatic text summaries [El-Kassas et al. 2021] Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric [Lin 2004].

Thus, the main contributions of this paper can be summarized as follows:

- The problem of generic extractive multi-document multi-language text summarization was formulated as an integer quadratic combinatorial optimization problem, involving the optimization of the following criteria: maximization of summary relevance, minimization of redundancy, and maximization of the informativeness of the selected sentences.
- HSSFLA, a memetic algorithm based on swarm intelligence, was developed to solve this problem for the first time.
- HSSFLA creates holistic summaries, in which it evaluates the quality of the summary as a whole, rather than focusing exhaustively on finding the best individual sentences.
- Experiments were conducted using the DUC2001 and DUC2002 datasets evaluating with ROUGE metrics.
- HSSFLA outperforms the results of the scientific literature in DUC2001 and DUC2002.

The remainder of this paper is organized as follows. Section 2 provides a brief literature review on document summarization methods based on Swarm Intelligence and Memetic Algorithms. Section 4 formulates the generic multi-document text summarization problem as an integer quadratic combinatorial optimization problem. Section 5 describes the fitness function to assess the quality of the candidate summaries generated by the HSSFLA algorithm. Section 6 introduces the basic Shuffled Frog-Algorithm (SFLA), describes in detail the Holistic Text Summarization with Shuffled Frog-Leaping Algorithm (HSSFLA), and the mutation strategy. Section 7 describes the preprocessed steps performed in the raw documents before executing the algorithm. Section 8 presents the datasets used, evaluation metrics, parameter settings, results obtained with the proposed approach, and comparisons with other methods in the literature. Finally, Section 9 concludes the paper and discusses directions for future research.

2. Related Work

This section provides a review of the main optimization techniques relevant to extractive text summarization based on Swarm Intelligence and Memetic Algorithms. For each study, the model used, the implemented algorithm, and the optimization objectives are presented. Additionally, the review is structured chronologically and categorized by classes of optimization methods.

Swarm intelligence-based approaches have been developed to address the text summarization problem. [Sanchez-Gomez et al. 2018] formulated text summarization as a multi-objective optimization problem and proposed the Multi-Objective Artificial Bee Colony (MOABC) algorithm, specifically designed for generic extractive summarization. This method simultaneously optimizes three critical criteria: summary length, content coverage, and redundancy reduction. Extending this work, [Sanchez-Gomez et al. 2020] introduced the Multi-Objective Artificial Bee Colony based on Decomposition (MOABC/D) to tackle the extractive multi-document summarization task. The authors implemented an asynchronous parallel version of MOABC/D to efficiently exploit multi-core computational environments. Similarly, [Tomer and Kumar 2022] proposed a swarm intelligence-based algorithm known as Firefly-based Text Summarization (FbTS) for multi-document summarization. This approach incorporates an innovative fitness function that combines three components: a document-topic relationship factor, a cohesion factor, and a readability factor. Distinct from traditional approaches, FbTS evaluates textual features through this composite fitness function rather than relying solely on cosine similarity.

In the context of memetic algorithms, [Mendoza et al. 2014] addressed the extractive single-document summarization problem by formulating it as a binary optimization task. They developed MA-SingleDocSum, an algorithm that integrates genetic operators with a guided local search mechanism. This memetic approach combines the global exploration capabilities of evolutionary algorithms with the intensification properties of local search. Furthermore, [Sanchez-Gomez et al. 2022] proposed the Multi-Objective Shuffled Frog-Leaping Algorithm (MOSFLA), a query-oriented extractive multi-document summarization technique based on multi-objective optimization principles. The effectiveness of MOSFLA was validated using the Text Analysis Conference (TAC) dataset, with performance assessed through the ROUGE evaluation metric. Experimental results demonstrated substantial improvements over baseline methods, achieving gains of 25.41%, 7.13%, and 30.22% in ROUGE-1, ROUGE-2, and ROUGE-SU4, respectively. Additionally, the MOSFLA algorithm was applied to the Topically Diverse Query Focus Summarization (TD-QFS) dataset, specifically composed of medical texts, as a case study to further assess its applicability.

All the reviewed approaches employed ROUGE metrics in their experiments, primarily ROUGE-1 and ROUGE-2. However, some studies also included other variations of ROUGE, such as ROUGE-S, ROUGE-SU, ROUGE-SU4, and ROUGE-L, in their performance evaluations. Additionally, most studies used a variation of the Document Understanding Conference (DUC) dataset, with DUC2002 being the most frequently employed, followed by DUC2001. Therefore, in this study and its experiments, the primary ROUGE-1 and ROUGE-2 metrics, along with the DUC2002 and DUC2001 datasets, were adopted for comparative purposes.

3. Problem Statement

This section introduces the generic approach to multi-document summarization. In existing literature, the most commonly employed technique to represent sentences is vector-based word methods. In this method, each sentence is represented as a vector of words and determines the similarity between sentences using specific criteria, such as cosine similarity. Next, we introduce the vector representation of sentences, the chosen similarity criterion.

3.1. Sentence Representation

Each sentence is encoded as a word vector. Let $T = \{t_1, t_2, \ldots, t_m\}$ represent the set of all distinct terms in the document collection D, where m indicates the total number of terms. Accordingly, each sentence s_i in D is represented as an m-dimensional vector, defined as $s_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, where $i = 1, 2, \ldots, n$ and n correspond to the total number of sentences. In this vector, each element reflects the weight of the term t_k within the sentence s_i. The term weight w_{ik} is calculated using the term-frequency inverse-sentence-frequency (tf-isf) scheme, following the formulation proposed by [Salton and Buckley 1988], and is mathematically expressed in Equation (1).

$$w_{ik} = tf_{ik} \cdot \log(n/n_k) \tag{1}$$

where tf_{ik} represents the frequency of term t_k in sentence s_i, and n_k denotes the number of sentences in D that contain t_k.

3.2. Cosine Similarity Measure

The cosine similarity measure quantifies the similarity between two sentences, s_i and s_j, within the document collection D. The computation of this metric is given by Equation 2:

$$\begin{gathered}\text{cosim}(s_i, s_j) = \frac{\sum_{k=1}^{m} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2} \cdot \sqrt{\sum_{k=1}^{m} w_{jk}^2}} \\ \text{for } i, j = 1, 2, \ldots, n\end{gathered} \tag{2}$$

where $\sum_{k=1}^{m} w_{ik} w_{jk}$ represents the dot product between the vectors s_i and s_j, with w_{ik} and w_{jk} denoting the weights of term k in sentences s_i and s_j, respectively. The expressions $\sqrt{\sum_{k=1}^{m} w_{ik}^2}$ and $\sqrt{\sum_{k=1}^{m} w_{jk}^2}$ correspond to the norms (or magnitudes) of vectors s_i and s_j, respectively. The norm reflects the magnitude or length of a vector within the vector space. By normalizing the dot product through the multiplication of these norms, the cosine similarity produces a normalized similarity score, which reflects the cosine of the angle between the two vectors, resulting in a value bounded between -1 and 1.

3.3. Informativeness

In order to enhance informativeness, entropy ($\gamma H(A)$) in Equation 4 is calculated based on the relevance vector corresponding to the selected sentences, which represents their probability distribution. The relevance vector is represented by capital alpha (A).

Entropy functions as a measure of the informational diversity within the selected set of sentences. A higher entropy value indicates a more balanced and heterogeneous distribution of relevance scores, suggesting that the summary comprises varied and informative content, with no single sentence disproportionately influencing the summary. This greater uncertainty reflects a broader coverage of relevant information [Khurana and Bhatnagar 2022].

By integrating entropy into the selection process, the method promotes a balanced choice of sentences with diverse relevance levels, thereby improving the overall informativeness of the summary. The entropy is computed in Eq. 3 according to [Shannon 1948].

$$h = -\sum p_i \cdot \log_2(p_i) \tag{3}$$

4. Mathematical formulation of the optimization problem

The optimization problem is formally defined as follows. Let $D = \{d_1, d_2, d_3, \ldots, d_N\}$ denote a document collection consisting in N documents. Alternatively, this collection can be represented as $D = \{s_1, s_2, s_3, \ldots, s_n\}$, where n corresponds to the total number of sentences extracted from all documents in the collection. The objective is to generate a summary S by selecting a subset of sentences from D ($S \subset D$), while satisfying the following criteria:

- **Maximizing relevance**. The summary should include sentences that are most relevant according to the central topics of the documents.
- **Minimizing redundancy**. The summary should avoid including sentences that are too similar to each other.
- **Informativeness**. The summary should contain sentences that provide the most informative content.
- **Length constraint**. The summary should have a predefined length L.

Relevance maximization focuses on selecting sentences that are highly representative relative to the center of the document, whereas redundancy reduction aims to avoid a summary that includes sentences with significant similarity within the candidate summary. In addition, informativeness ensures that the selected set of sentences collectively contributes diverse and relevant information. To address these requirements, the objective function proposed by [Alguliyev et al. 2015] was adapted and enhanced to improve the selection of informative sentences by incorporating entropy maximization over the relevance values (α). Consequently, the following objective function was defined:

$$\text{maximize} \quad f(X) = \text{max\_rel} \cdot \text{min\_red} + \gamma H(A) \tag{4}$$

$$\text{subject to} \sum_{i=1}^{n} l_i x_i \leq L, \quad x_i \in \{0, 1\}, \text{ for } i = 1, \ldots, n \tag{5}$$

Where l_i represents the length of sentence S_i. The number of words measures both the size of the summary and the sentence. The terms that maximize the relevance of the sentences (max_rel), minimize the selection of redundant sentences (min_red) and promote informativenes are formulated as follows:

$$\text{max\_rel} = \left(\sum_{i=1}^{n} \alpha_i x_i \right) \tag{6}$$

$$\text{min\_red} = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{cosim}(s_i, s_j) \cdot x_i x_j + \epsilon} \tag{7}$$

The relative relevance (α) of the candidate summary sentences was computed as follows:

$$\alpha_i = \frac{\text{sim}(s_i, O)}{\sum_{j=1}^{n} \text{sim}(s_j, O)}, \text{i = 1, 2, ..., n} \tag{8}$$

where O is the center of the collection $D = \{s_1, s_2, ..., s_n\}$, and the k-th coordinate o_k of the center is calculated as:

$$o_k = \frac{1}{n} \sum_{i=1}^{n} w_{i,k} \tag{9}$$

The weight defined in Equation (8) determines the relative relevance of sentence s_i concerning the main content of the collection D.

5. Fitness Computation and Constraint Handling

A fitness function is defined to assess the quality of a candidate summary, as its value serves as an indicator of how well the summary addresses the optimization problem.

A penalty term was incorporated into the fitness function to transform the constrained problem into an unconstrained one.

An additional penalty term is introduced [Alguliyev et al. 2015] to discourage infeasible solutions by applying a penalty factor $\beta(\beta > 0)$. In addition, the proposed fitness function used a progressive (quadratic) penalty, allowing a smoother exploration near the length limit and severely penalizing solutions that significantly exceed it.

The fitness function is formally defined as follows:

$$fit(X) = f(X) \cdot \exp\left(-\beta \cdot \left(\max\left(0, \sum_{i=1}^{n} l_i \cdot x_i - L\right)\right)^2\right) \tag{10}$$

6. Holistic Text Summarization with Shuffled Frog-Leaping Algorithm

In this section, the Holistic Text Summarization with Shuffled Frog-Leaping Algorithm (HSSFLA) is presented. First, the basic SFLA algorithm is described. Then, the preprocessing steps are defined. Finally, the main steps of HSSFLA and its core operators are explained.

6.1. Basic algorithm

The memetic metaheuristic Shuffled Frog-Leaping Algorithm (SFLA), introduced by [Eusuff et al. 2006], was designed to address combinatorial optimization problems. SFLA is a population-based cooperative search method inspired by the concept of natural memetics. The algorithm operates by dividing the population of candidate solutions (frogs) into groups known as memeplexes, within which individuals interact and share information. In this context, the virtual frogs serve as carriers of memes, where a meme represents a unit of cultural evolution. The algorithm conducts independent local searches within each memeplex while simultaneously performing global exploration by periodically shuffling the frogs and reorganizing them into new memeplexes. The detailed procedure of the SFLA is outlined in Algorithm 1.

Algorithm 1 Basic SFLA pseudocode

```
$Pop \leftarrow init\_population(popsize)$
$Pop \leftarrow calculate\_fitness(Pop)$
$Pop \leftarrow sort\_by\_fitness(Pop)$
while not stop_criteria do
    $Memplexes \leftarrow divide\_pop\_into\_memplexes(Pop, num)$
    $Memplexes \leftarrow local\_search(Memplexes, num, improvsmax)$
    $Pop \leftarrow combine\_evolved\_memplexes(Memplexes)$
    $Pop \leftarrow calculate\_fitness(Pop)$
    $Pop \leftarrow sort\_by\_fitness(Pop)$
end while=0
```

6.2. Main Steps of Topic HSSFLA

The SFLA metaheuristic was selected as the foundation for addressing the generic extractive summarization problem due to the low complexity involved in adapting it to a generic summarization approach (HSSFLA). In addition, it has the advantage of performing a local search and a global search to optimize the objective function (Equation 4) in a relatively simple way. The detailed HSSFLA algorithm is described in Algorithm 2. In addition, the code with the implementation is available in a public repository on the GitHub[1] platform.

6.3. Mutation

Since the method presented in this study is a generic summarization approach, all comparisons are performed with the centroid rather than with the query, as originally intended in [Sanchez-Gomez et al. 2022]. This section describe the mutation strategy proposed by [Sanchez-Gomez et al. 2022] and adapted for the proposed HSSFLA.

The mutation process consists of modifying a candidate summary (individual) by performing one of three operations: addition (Eq. 11), removal (Eq. 12), or replacement of a sentence, depending on the selected mutation type. Each mutation type has an equal probability of being selected, with only one applied during each mutation event. The mutation probability is defined as $p_m = 1/n$, where n denotes the total number of sentences, ensuring that precisely one sentence is altered per iteration. Mutation is applied unconditionally, regardless of whether the modified sentence improves the solution quality.

The addition operation involves incorporating a sentence from the set of available sentences that is not currently part of the candidate summary. The selected sentence ($s_i \notin S$) is expected to improve the quality of the summary. Specifically, the cosine similarity between the new sentence and the centroid must exceed the average cosine similarity of the existing sentences in the candidate summary relative to the centroid:

$$cosim(s_i, O) > \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, O). \tag{11}$$

A sentence $s_i \notin S$ is randomly selected from the document set D and is added to the candidate summary if it satisfies the specified condition. If the condition is not met,

[1] https://github.com/k3ybladewielder/hssfla

Algorithm 2 HSSFLA pseudocode

```
$Population \leftarrow init\_population(popsize)$
$Population \leftarrow calculate\_objective\_functions(Population, popsize)$
$Population \leftarrow sort\_by\_fitness(Population, popsize)$
for $cycle = 1$ to $cyclesmax$ do
    $X_{bestG} \leftarrow select\_best\_global(Population)$
    $Memplexes \leftarrow divide\_pop\_into\_memplexes(Population, memnum)$
    for $m = 1$ to $memesnum$ do
        for $i = 1$ to $improvsmax$ do
            $X_{bestL} \leftarrow select\_best\_local(Memplexes[m])$
            $X_{worstL} \leftarrow select\_worst\_local(Memplexes[m])$
            $save\_worst\_local(Population, X_{worstL})$
            $X_{new} \leftarrow mutate\_solution(X_{bestL}, p_m)$
            if $X_{new} \succ X_{worstL}$ then
                $save\_solution(Memplexes[m], X_{new})$
            else
                $X_{new} \leftarrow mutate\_solution(X_{bestG}, p_m)$
                if $X_{new} \succ X_{worstL}$ then
                    $save\_solution(Memplexes[m], X_{new})$
                else
                    $X_{new} \leftarrow random\_solution()$
                    $save\_solution(Memplexes[m], X_{new})$
                end if
            end if
            $Memplexes[m] \leftarrow sort\_by\_dominance(Memplexes[m])$
        end for
    end for
    $Population \leftarrow combine\_evolved\_memplexes(Memplexes)$
    $Population \leftarrow calculate\_objective\_functions(Population, popsize * 2)$
    $Population \leftarrow sort\_by\_fitness(Population, popsize * 2)$
    $Savebest\_individual, best\_fitness, best\_sentences$
end for=0
```

the sentence $s_i \notin S$ with the highest cosine similarity to the centroid is selected and added instead.

The sentence removal operation deletes one sentence from the candidate summary. To ensure that the removal does not negatively affect the summary's quality, the cosine similarity between the sentence $s_i \in S$ and the centroid must be lower than the average cosine similarity of all sentences currently in the summary relative to the centroid:

$$cosim(s_i, O) < \frac{1}{n}\sum_{j=1}^{n} cosim(s_j, O). \tag{12}$$

Similar to the addition process, a sentence $s_i \in S$ is randomly selected from the candidate summary. If it satisfies the specified condition, it is removed; otherwise, subsequent sentences $s_i \in S$ are evaluated until the condition is fulfilled. If no sentence satisfies the removal criterion, the sentence $s_i \in S$ with the lowest cosine similarity to the centroid is removed.

The sentence replacement operation involves exchanging a sentence from the document collection D that is not currently part of the summary with one that is already included. This mutation process simultaneously removes a sentence from the summary and incorporates a different sentence from the document collection.

7. Preprocessing

Before executing the algorithm, some preprocessing steps must be performed on the documents in the collection D. These steps are as follows:

1. **Segmentation**. Each sentence in the document must be extracted individually, clearly defining its beginning and end.
2. **Tokenization**. All words in a sentence are tokenized to remove special characters, question marks, and punctuation.
3. **Stopword Removal**. Words that do not carry significant semantic meaning and appear frequently, such as prepositions, conjunctions, and articles, are excluded from the sentences. The stopword removal operation was performed using the NLTK library[2].
4. **Stemming**. The root of each word was extracted using the SnowballStemmer module from the NLTK library.
5. **Representation**. Words in the sentences are numerically represented through the application of TF-ISF.

8. Experimental results

This section describes the datasets DUC2001 and DUC2002 used, the performance evaluation metric, the parameter settings process, the results obtained with the proposed approach, and a comparison with results from the literature.

[2] https://www.nltk.org/

8.1. Datasets

The performance of the proposed algorithm was assessed using the DUC2001 and DUC2002 datasets, which are publicly available benchmark corpora released by the Document Understanding Conference (DUC) [DUC 2024] for evaluating automatic text summarization systems. These datasets comprise English news articles spanning a diverse range of topics, sourced from outlets such as the Financial Times, Associated Press, and The Wall Street Journal. Specifically, DUC2001 includes 30 topics with a total of 302 documents, whereas DUC2002 contains 59 topics comprising 533 documents. Both datasets provide reference summaries, each limited to 100 words, generated manually by human experts for every document. Each instance in the datasets consists of the original document paired with its corresponding human-written summary, which serves as the gold standard for evaluation. A detailed overview of the DUC2001 and DUC2002 datasets is presented in Table 1.

Table 1. Description of the DUC2001 and DUC2002 datasets.

	DUC2001	DUC2002
No. of Topics	30	59
No. of Documents	302	533
Average No. of Sentences per Topic	59.5	150.55
Average No. of Sentences per Document	5.91	15.55
Average No. of Terms per Topic	1012.97	2138.68
Average No. of Terms per Document	100.62	236.74

On average, the reference summaries for the DUC2002 dataset are longer, both in terms of the number of sentences per topic and per document, as well as the number of words per topic and per document. All documents were segmented into sentences and preprocessed. The data preprocessing process is described in Section 7.

8.2. Performance Evaluation Metrics

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric was adopted as the evaluation measure, as it is widely used in the literature for assessing automatic text summarization tasks. ROUGE compares the generated summary with the human-made reference summary (gold standard) [El-Kassas et al. 2021]. It measures summary quality by counting the overlap of N-gram units, word sequences, and word pairs between the automatic summary and the reference summary.

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{RefSummaries}} \sum_{\text{N-grams} \in S} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{S \in \text{RefSummaries}} \sum_{\text{N-grams} \in S} \text{Count}(\text{N-gram})} \tag{13}$$

In the following experiments, we report the recall values for ROUGE-1 and ROUGE-2 using the standard ROUGE (Equation 13).

8.3. Parameter Settings

To define the parameters, a test was conducted using a sample from DUC2001. The sample was determined with a 0.95 confidence level in the sample size calculation, resulting in 169 (55%) examples. The test involved running the algorithm for 5, 10 and 15 cycles (*cyclesmax*), randomly selecting values for the number of iterations (*improvsmax*) and the number of memeplexes (*memesnum*), the gamma (γ), the population siza (*pop_size*), and measuring its performance using the metrics described in Subsection 8.2. The possible values for the number of iterations (*improvsmax*) were 25, 50, 75, and 100, for the number of memeplexes (*memesnum*) were set to 5, 10, 15, 20, and 25, for gamma (γ) were 0.25, 0.5, 0.75 and 0.9, while the possible values for the population size were 100 (*pop_size*) 100, 200, 300, 400 and 500. The test was performed independently 31 times for each cycle parameter (*cyclesmax*), and in each test execution, the parameters were selected randomly, with each having an equal probability of being chosen.

After conducting 31 executions for each document in the calculated sample, a total of 92 parameter configurations were generated, selected randomly. To identify the best parameters, the 20 configurations with the highest average ROUGE-1 score were selected and stored. Subsequently, the 20 configurations with the highest ROUGE-1 average F-measure were also selected. Both groups were combined into a list without duplicate configurations, resulting in 20 unique configurations.

Next, a one-way ANOVA hypothesis test was applied using the *Scipy*[3] library to determine whether any configuration achieved a significantly higher mean result compared to the others. One-way ANOVA tests the null hypothesis that two or more groups share the same population mean. When applied to the entire set of parameter configurations, using ROUGE-1 to form the groups, the test yielded an F-statistic of 0.35645 and a p-value of 0.99999. When applied to the top 20 configurations, the results were an F-statistic of 0.2414 and a p-value of 0.9996.

The low F-statistic values in both cases indicate that the variability between the group means is very small compared to the variability within each group. In other words, the means across the groups are highly similar. In both scenarios, with a significance level α set at 0.05, the high p-value ($p > \alpha$) suggests that the probability of observing no differences in the sample data is substantially high, assuming the null hypothesis holds true. Consequently, there is insufficient statistical evidence to reject the null hypothesis that any sample mean of a given configuration (whether among all configurations or among the top 20) exhibits a statistically significant difference.

Therefore, to select the parameter configuration used in the experiments with the DUC2001 and DUC2002 datasets, a weighted evaluation of the ROUGE metrics was performed. The weights assigned were 0.4 for both Recall and F-measure of ROUGE-1, and 0.1 for the F-measure of ROUGE-2. Each metric was multiplied by its respective weight, and the resulting values were summed to compute a weighted score. Table 2 presents the results for each configuration, highlighting the one with the highest weighted score. In the **Config** column, the values correspond, respectively, to the parameters *cyclesmax*, *pop_size*, *improvsmax*, *memesnum*, and *gamma*.

8.4. Results with the proposed approach

In this subsection, the results obtained performing HSSFLA on DUC2001 and DUC2002. A statistical analysis was performed on the ROUGE-1 and ROUGE-2 scores for all topics across the datasets used.

[3] https://scipy.org/

Table 2. Best 20 results performed on DUC2001 with differents configurations and it's ROUGE evaluation metrics.

Config	ROUGE-1 R	ROUGE-2 R	ROUGE-1 F1	ROUGE-2 F1	Weighted
10-200-25-5-0.5	0.895009	0.941808	0.933316	0.934096	0.918920
5-200-50-20-0.5	0.836247	0.917020	0.881764	1.000000	0.878906
15-300-100-5-0.9	1.000000	1.000000	0.728359	0.816262	0.872970
5-400-100-25-0.5	0.828825	0.957269	0.771117	0.951900	0.830894
10-300-25-5-0.9	0.634915	0.505620	1.000000	0.823915	0.786919
10-200-25-15-0.9	0.776983	0.956226	0.655743	0.949577	0.763670
15-200-25-15-0.5	0.820917	0.841579	0.666224	0.827401	0.761755
15-400-100-10-0.9	0.704377	0.828306	0.735278	0.831737	0.741866
10-500-25-15-0.25	0.901444	0.901595	0.554950	0.647784	0.737495
10-500-25-10-0.25	0.764803	0.680823	0.726792	0.658799	0.730600
5-200-50-10-0.9	0.660885	0.694636	0.787702	0.746091	0.723507
5-200-100-5-0.75	0.751163	0.774060	0.674126	0.729329	0.720455
5-400-100-5-0.5	0.638754	0.728994	0.763853	0.815231	0.715466
5-100-75-20-0.25	0.556204	0.824279	0.693786	0.960909	0.678515
10-100-50-10-0.5	0.727246	0.673544	0.652443	0.568588	0.676088
5-300-75-5-0.25	0.592966	0.525563	0.754388	0.743605	0.665858
10-300-75-5-0.75	0.569153	0.695551	0.701375	0.811051	0.658871
15-500-75-10-0.25	0.681431	0.868207	0.508022	0.843672	0.646969
15-100-25-5-0.75	0.701763	0.801257	0.540416	0.668539	0.643851
15-300-25-10-0.75	0.710391	0.708510	0.575787	0.561802	0.641503

The tables include the mean, median, and standard deviation, as well as the first and third quartiles (Q1 and Q3), along with the minimum and maximum values for the ROUGE scores. These values were computed based on 31 independent repetitions per sample for both the DUC2001 and DUC2002 datasets. Tables 3 and 4 describe the results.

Table 3. Results obtained by HSSFLA for ROUGE-1 and ROUGE-2 using DUC2001

HSSFLA	ROUGE-1 Precision	ROUGE-2 Precision	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-1 F1	ROUGE-2 F1
Mean	0.187162	0.082696	0.594694	0.265815	0.257536	0.116835
Median	0.168359	0.064916	0.626866	0.253521	0.251121	0.100840
Standard Deviation	0.102535	0.077730	0.194115	0.153408	0.109598	0.083058
Q1	0.119116	0.036649	0.514706	0.161765	0.186880	0.059406
Q3	0.232759	0.107289	0.722222	0.354839	0.325792	0.157521
Minimum	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Maximum	1.000000	1.000000	1.000000	0.956522	0.769231	0.695652

Table 4. Results obtained by HSSFLA for ROUGE-1 and ROUGE-2 using DUC2002

HSSFLA	ROUGE-1 Precision	ROUGE-2 Precision	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-1 F1	ROUGE-2 F1
Mean	0.236632	0.111303	0.575184	0.263056	0.304667	0.140688
Median	0.209302	0.087591	0.600000	0.250000	0.299270	0.126214
Standard Deviation	0.128821	0.107376	0.188940	0.143518	0.112938	0.086880
Q1	0.156425	0.053435	0.485294	0.163636	0.234604	0.080460
Q3	0.287770	0.138554	0.701493	0.352941	0.375000	0.186916
Minimum	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Maximum	1.000000	1.000000	1.000000	0.936508	0.857143	0.725664

The average ROUGE-N represents the expected value or the mean performance of the model across all evaluated samples. The median indicates the central value of the distribution, meaning that 50% of the summaries achieved performance above and 50% below this value. The median is less sensitive to extreme values (outliers) than the mean, offering a robust measure of typical performance. The first quartile (Q1) represents the value below which 25% of the results are located. In the context of ROUGE, Q1 reflects the performance in the worst 25% of cases, helping to assess the method's consistency and stability under less favorable scenarios. The third quartile (Q3) represents the value below which 75% of the results fall, meaning it encompasses most of the results except the top 25%. Q3 contributes to understanding the performance variability, indicating how well the method performs for the majority of cases.

The mean ROUGE-1 Precision increased from 0.187 in DUC2001 to 0.237 in DUC2002, and ROUGE-2 Precision rose from 0.083 to 0.111. This suggests that the model was able to generate summaries with higher accuracy, that is, with a greater proportion of generated n-grams that actually appear in the reference summaries of the DUC2002 dataset. The median and quartile values confirm this trend, showing consistent increases across these parameters, which indicates improvements in both typical performance and result stability.

The average recall for ROUGE-1 showed a slight decrease from 0.595 to 0.575, while ROUGE-2 recall remained almost unchanged (from 0.266 to 0.263). Similarly, the median and quartiles for recall exhibit minor variations, suggesting that the model retrieves a similar proportion of relevant n-grams in both datasets. The slight decrease in ROUGE-1 average recall may indicate that, despite achieving higher precision, the method retrieved a smaller fraction of the relevant content, possibly reflecting a more conservative sentence selection strategy.

The F1-score, which combines precision and recall, shows average improvements from 0.258 to 0.305 for ROUGE-1 and from 0.117 to 0.141 for ROUGE-2 when moving from DUC2001 to DUC2002. This increase indicates a more favorable balance between precision and recall in the second dataset, further supported by higher median and quartile values.

The first and third quartile values for precision and F1-score are also higher in DUC2002, indicating that both the worst and best 25% of results improved compared to DUC2001. However, recall shows a slight reduction at Q1 for ROUGE-1, suggesting greater variability in lower-performance cases.

Figures 1 and 2 include boxplots and histograms of the results obtained for ROUGE-1 and ROUGE-2.

In conclusion, the analysis of descriptive statistics reveals that the HSSFLA method demonstrates slightly superior performance in terms of precision and F1-score on the DUC2002 dataset compared to DUC2001. This indicates an increased ability of the method to produce summaries that are more faithful to the reference documents, even though the parameters were optimized exclusively with samples from the DUC2001. On the other hand, recall remains relatively stable or slightly reduced, suggesting a potential prioritization of precision over complete retrieval of relevant content. The stability of the quartile values reinforces the consistency of this behavior across different performance levels, highlighting the robustness of the method in the DUC2002 scenario.

8.5. Comparison with results from other approaches

The following subsection presents the results obtained by other approaches, which will be compared with the results of the proposed methodology in this study.

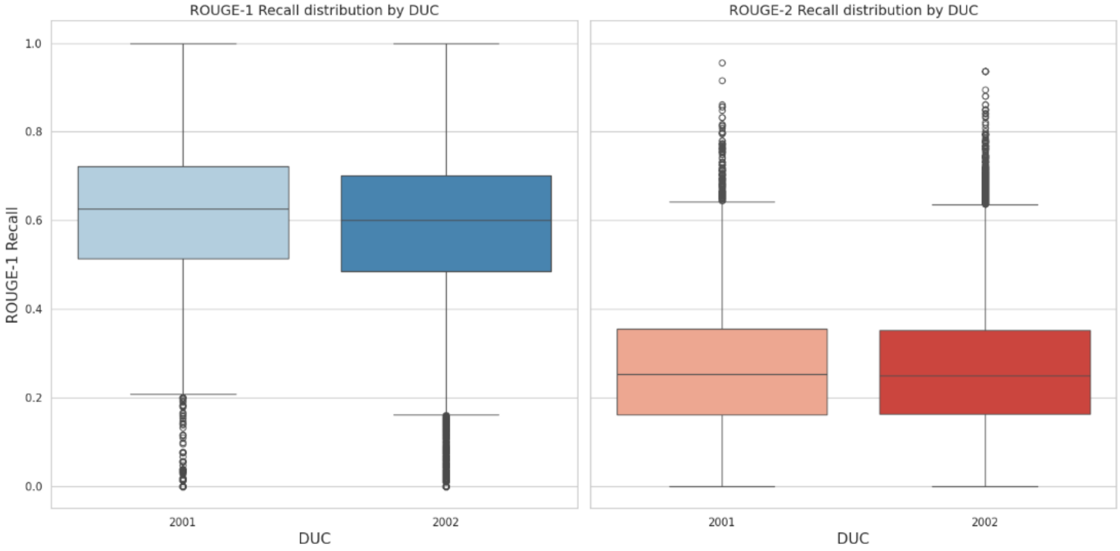

Figure 1. Boxplots obtained by HSSFLA for ROUGE-1, ROUGE-2 (Recall values).

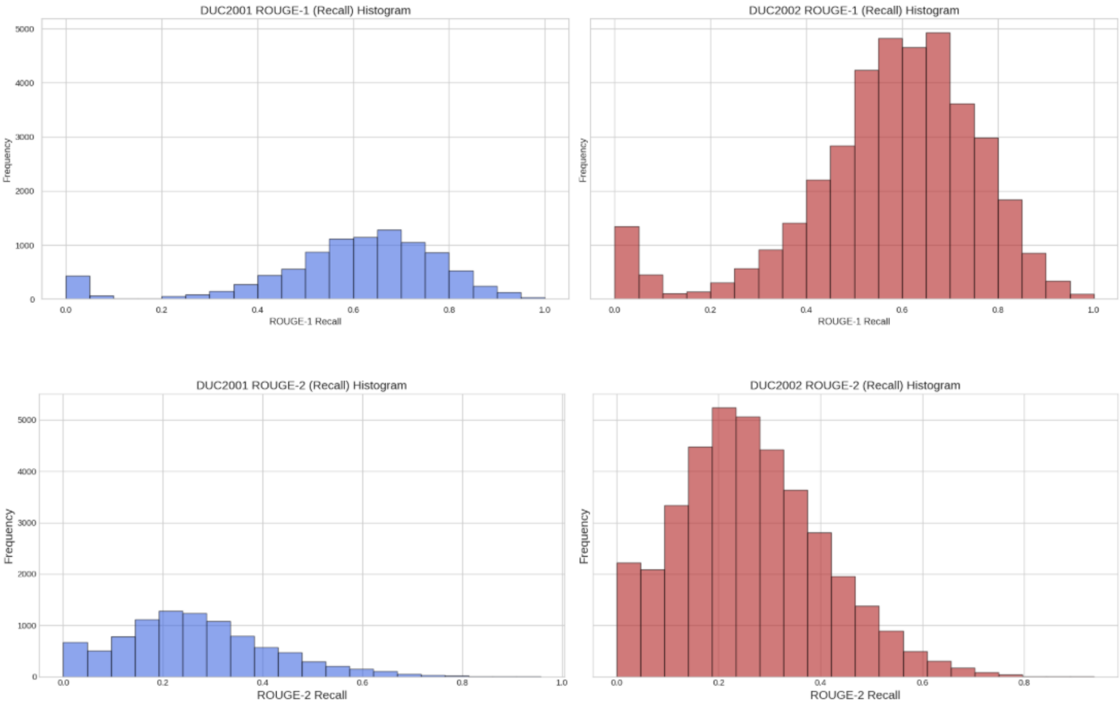

Figure 2. Histograms obtained by HSSFLA for ROUGE-1 and ROUGE-2 (Recall values).

Tables 5 and 6 show the comparative results for the DUC2001 and DUC2002 datasets based on the evaluated metrics for each competing method. They report the average Recall score for ROUGE-1 and ROUGE-2, as well as (in brackets) the percentage improvement achieved by HSSFLA. The symbol ”-” is used when the result for a given metric is unavailable. The percentage is calculated as $((\text{proposed metric} - \text{baseline metric})/\text{baseline metric}) * 100$

For DUC2001, the state-of-the-art methods used for comparison include Branch

Table 5. Comparison of the proposed model with other methods using ROUGE-1 and ROUGE-2 on DUC2001 (Recall Values). The HSSFLA and best metric are highlighted in bold.

Methods	ROUGE-1		ROUGE-2	
HSSFLA	**0.594694**	-	**0.265815**	-
FEC_B&B	**0.4971**	(+19.63)	**0.2112**	(+25.86)
FEC_Gap	0.4865	(+22.24)	0.2081	(+27.73)
FEC_GAP*	0.4816	(+23.48)	0.2054	(+29.41)
FEC without Gap	0.4632	(+28.39)	0.1873	(+41.92)
FEC_WGAP	0.4621	(+28.69)	0.1866	(+42.45)
FEC_WGAP*	0.4613	(+28.92)	0.1831	(+45.17)
Average Result	0.4754	(+25.12)	0.1971	(+34.91)

Table 6. Comparison of the proposed model with other methods using ROUGE-1 and ROUGE-2 on DUC2002 (Recall Values). The HSSFLA and best metric are highlighted in bold.

Methods	ROUGE-1		ROUGE-2	
HSSFLA	**0.575184**	-	**0.263056**	-
FEC_B&B	**0.4987**	(+15.34)	**0.2187**	(+20.28)
FEC without Gap	0.4637	(+24.04)	0.1889	(+39.26)
FEC_WGAP	0.4636	(+24.07)	0.2138	(+23.04)
FEC_GAP*	0.4634	(+24.12)	0.1966	(+33.80)
FEC_Gap	0.4613	(+24.69)	0.2163	(+21.62)
FEC_WGAP*	0.4613	(+24.69)	0.1882	(+39.77)
Punctuation+Root Stemming	0.4572	(+25.81)	-	-
FbTS	0.4380	(+31.31)	0.2121	(+24.01)
Punctuation+Lemma Stemming	0.352	(+63.40)	-	-
FUZZY CST with COM	0.33206	(+73.22)	0.12806	(+105.42)
Average Result	0.4391	(+35.42)	0.1953	(+36.08)

and Bound (B&B), standard gap statistics (Gap), gap statistics without logarithmic function (Gap*), weighted gap statistics (WGap), and weighted gap statistics without logarithmic function (WGap*), as proposed by [Verma et al. 2022] are compared with HSSFLA.

For DUC2002, the comparisons were made against the methods FbTS [Tomer and Kumar 2022], FUZZY CST with COM [Kumar et al. 2014], FEC without Gap [Verma et al. 2022], FEC_B&B [Verma et al. 2022], FEC_Gap [Verma et al. 2022], FEC_GAP* [Verma et al. 2022], FEC_WGAP [Verma et al. 2022], FEC_WGAP* [Verma et al. 2022], Punctuation+Lemma Stemming [Alqaisi et al. 2020], and Punctuation+Root Stemming [Alqaisi et al. 2020].

The proposed HSSFLA outperformed state-of-the-art approaches in both DUC2001 and DUC2002 benchmarks. For DUC2001, it achieved an increase of 19.63% and 25.86% in the average Recall for ROUGE-1 and ROUGE-2, respectively, compared to the best existing method. When compared to the worst-performing method, the improvement was even more significant, reaching 28.92% for ROUGE-1 and 45.17% for ROUGE-2.

For DUC2002, HSSFLA achieved a gain of 15.34% and 20.28% in ROUGE-1 and ROUGE-2, respectively, relative to the best competing method. Compared to the weakest methods, the improvement was even more substantial, reaching 73.22% for ROUGE-1 and an impressive 105.42% for ROUGE-2.

On average, the proposed approach resulted in a percentage improvement of 25.12% in ROUGE-1 and 34.91% in ROUGE-2 for the DUC2001 dataset. Meanwhile, for DUC2002, it achieved an average percentage improvement of 35.42% in ROUGE-1 and 36.08% in ROUGE-2.

9. Conclusion and future work

We present an unsupervised, optimization-based approach for automatic text summarization. In the proposed method, text summarization is formulated as an integer quadratic combinatorial optimization problem. The criteria to be optimized include maximizing the relevance of the selected candidate summary, minimizing redundancy by promoting diversity, maximizing the informativeness of the selected sentences, and ensuring compliance with a maximum summary length constraint. The proposed approach is applicable to both single-document and multi-document summarization tasks.

In this paper, a memetic algorithm, Holistic Text Summarization with Shuffle Frog-Leaping Algorithm (HSSLF), was designed, implemented, and developed for the first time to solve this problem. HSSFLA is a population-based swarm intelligence algorithm that introduces a new optimization criterion (informativeness), along with a mutation operator specifically adapted to the generic summarization problem. In HSSFLA, the exploitation of the best solutions (local search) is performed within memeplexes (candidate solutions). Additionally, candidate solutions are periodically shuffled and reorganized into new memeplexes to ensure a global search.

Experiments were conducted using the DUC2001 and DUC2002 benchmark datasets. After performing a statistical analysis on 835 documents, the results indicate that HSSFLA provides superior outcomes compared to other bioinspired-based approaches in the scientific literature. A total of 16 methods from other authors were used for comparison. HSSFLA achieved an average percentage improvement of 25.12% in ROUGE-1 and 34.918% in ROUGE-2 on the DUC2001 dataset. Meanwhile, for DUC2002, it achieved an average improvement of 35.424% in ROUGE-1 and 36.08% in ROUGE-2. The evaluation metric of interest for both cases was recall.

For future research, we intend to further investigate the optimization problem by exploring other sentence representation methods, like embedding-based methods, and expand the benchmarking comparing to other types of algorithms like deep learning, machine learning and computational linguistic summarization based approaches. Also, optimize the objective function using novel mutation strategies and employing different swarm intelligence algorithms, incorporating diverse mutation strategies and comparing their results.

10. Limitations

Despite the promising results obtained, this study presents limitations that should be acknowledged.

First, the sentence representation is based on the traditional TF-IDF weighting scheme, which, although effective, does not capture semantic relationships as well as more recent embedding-based approaches, such as those using transformer architectures. This choice may limit the expressiveness and contextual understanding of the sentence vectors used in similarity calculations.

Second, the parameter configuration was derived exclusively from evaluations on the DUC2001 dataset, which may have the potential to introduce dataset-specific biases and reduce the generalizability of the results to other corpora. Additionally, the performance evaluation was restricted to comparisons with bio-inspired models only, excluding all non-bio-inspired state-of-the-art extractive and abstractive summarization methods, which limits the contextual benchmarking of the proposed approach.

Lastly, the assessment was based solely on the ROUGE metric, particularly recall-oriented variants, without incorporating complementary evaluation metrics that account for aspects such as precision, fluency, or semantic adequacy. These limitations suggest avenues for future work aimed at enhancing the model's robustness, generalizability, and comparative relevance.

Acknowledgments

During the preparation of this work, the author(s) used the generative artificial intelligence tool ChatGPT [ChatGPT 2025] only for spell-checking. After using this tool/service, the author(s) reviewed and edited the content as needed and assume full responsibility for the content of the published article.

References

Abbasi-ghalehtaki, R., Khotanlou, H., and Esmaeilpour, M. (2016). Fuzzy evolutionary cellular learning automata model for text summarization. *Swarm and Evolutionary Computation*, 30:11 – 26. Cited by: 52.

Alguliev, R. M., Aliguliyev, R. M., and Isazade, N. R. (2012). Desamc+docsum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization. *Knowledge-Based Systems*, 36:21 – 38. Cited by: 56.

Alguliyev, R. M., Aliguliyev, R. M., and Isazade, N. R. (2015). An unsupervised approach to generating generic summaries of documents. *Applied Soft Computing Journal*, 34:236 – 250. Cited by: 44.

Alqaisi, R., Ghanem, W., and Qaroush, A. (2020). Extractive multi-document arabic text summarization using evolutionary multi-objective optimization with k-medoid clustering. *IEEE Access*, 8:228206 – 228224. Cited by: 34; All Open Access, Gold Open Access.

ChatGPT (2025). Chatgpt. Disponível em: https://chat.openai.com/. Acesso em: janeiro de 2025.

DUC (2024). Document understanding conference.

El-Kassas, W. S., Salama, C. R., Rafea, A. A., and Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Eusuff, M., Lansey, K., and Pasha, F. (2006). Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2):129–154. Published online: 25 Jan 2007, Received: 29 Sep 2004.

Gomes, L. and Oliveira, H. (2019). A multi-document summarization system for news articles in portuguese using integer linear programming. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pages 622–633, Porto Alegre, RS, Brasil. SBC.

Huang, L., He, Y., Wei, F., and Li, W. (2010). Modeling document summarization as multi-objective optimization. page 382 – 386. Cited by: 42.

Jorge, G. A. Z., Bezerra, D. A., Xavier, C. C., and Pardo, T. A. S. (2025). Multilingual extractive summarization: Investigating state-of-the-art methods for english and brazilian portuguese. In Paes, A. and Verri, F. A. N., editors, *Intelligent Systems*, pages 212–223, Cham. Springer Nature Switzerland.

Khurana, A. and Bhatnagar, V. (2022). Investigating entropy for extractive document summarization. *Expert Systems with Applications*, 187:115820.

Kumar, Y. J., Salim, N., Abuobieda, A., and Albaham, A. T. (2014). Multi document summarization based on news components using fuzzy cross-document relations. *Applied Soft Computing Journal*, 21:265 – 279. Cited by: 34.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., and León, E. (2014). Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications*, 41(9):4158 – 4169. Cited by: 114.

Saini, N., Saha, S., Jangra, A., and Bhattacharyya, P. (2019). Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm. *Knowledge-Based Systems*, 164:45 – 67. Cited by: 61.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2018). Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. *Knowledge-Based Systems*, 159:1 – 8. Cited by: 84.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2020). A decomposition-based multi-objective optimization approach for extractive multi-document text summarization. *Applied Soft Computing Journal*, 91. Cited by: 39.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2022). A multi-objective memetic algorithm for query-oriented text summarization: Medicine texts as a case study. *Expert Systems with Applications*, 198:116769.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2024). An indicator-based multi-objective variable neighborhood search approach for query-focused summarization. *Swarm and Evolutionary Computation*, 91. Cited by: 0.

Sarmento, M. and de Oliveira, H. (2024). Sumarização automática de artigos de notícias em português: Da extração à abstração com abordagens clássicas e modelos de neurais. In *Anais do XV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 139–148, Porto Alegre, RS, Brasil. SBC.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Tomer, M. and Kumar, M. (2022). Multi-document extractive text summarization based on firefly algorithm. *Journal of King Saud University - Computer and Information Sciences*, 34(8, Part B):6057–6065.

Verma, P., Verma, A., and Pal, S. (2022). An approach for extractive text summarization using fuzzy evolutionary and clustering algorithms. *Applied Soft Computing*, 120. Cited by: 33.