MuSEvo: Assessing the Robustness of Multilingual Summarization Metrics Through Evolutionary Adversarial Natural Language Processing-Based Attacks

Gema Bello-Orgaz[1], Aidan Dakhama[2], Cristian Ramírez-Atencia[1], and Hector D. Menendez[3(✉)]

[1] Universidad Politecnica de Madrid, Carretera de Valencia (A3) km. 7, 28031 Madrid, Spain
{gema.borgaz,cristian.ramirez}@upm.es
[2] University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, Scotland, UK
adakhama@ed.ac.uk
[3] King's College London, 30 Aldwych, London WC2B 4BG, UK
hector.menendez@kcl.ac.uk

Abstract. This paper investigates the robustness of multilingual summarization metrics through adversarial Natural Language Processing (NLP) attacks. We evaluate metric resilience across three languages using a multi-objective evolutionary algorithm. The initial population is derived from the original text's tokens, which are stochastically replaced by synonyms from a dynamically generated multilingual lexicon. To ensure the accuracy of the adversarial examples, we implement a correction step with a rule-based grammar and style checker that validates the syntactic and semantic coherence of the perturbed text. Conducted on the MuSE framework, this study rigorously analyzes the sensitivity of conventional metrics to semantic-preserving variations, contributing to the development of more reliable and resilient evaluation systems for multilingual summarization.

Keywords: Summarization · Multi-lingua · Large Language Models · Evolutionary Algorithms · Multi-Objective Optimization · Natural Language Processing · Adversarial Attacks · MuSE

1 Introduction

The problem of automatic text summarization is typically divided into two main approaches [12]. The first methodology is Extractive Summarization [14], which selects specific sentences or parts from the original text it compiles to create the final summary. In contrast, Abstractive Summarization synthesizes a full summary by understanding the original text and its meaning, while retaining its key facts [17]. The goal of both techniques is to provide the same information condensed.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 381–399, 2027.
https://doi.org/10.1007/978-3-032-30488-9_24

Both summarization techniques have been extensively used in the literature, yet evaluating summaries remains an open problem in the field [6]. The main challenge for evaluation metrics, originally designed for extractive summarization, is to deal with the infinite number of equivalent summaries that can be generated through abstractive summarization. Quality metrics evaluate the similarity between the original text and the summary, or the similarity between a human-provided text and the summary, normally employing n-gram methods that measure overlapping [6]. This approach would work if only one summary is valid; however, under multiple summaries, the quality can be adversarially targeted. Therefore, we need to identify new metrics that can represent the level of information retained from the original content.

If we look at the field of adversarial summarization and sequence-to-sequence tasks, we can see that the problem is even more complex. Adversarial methods have shown that by exploiting the output space of potential summaries, it is possible to control quality metrics without altering the generative content [22]. Normally, efforts to improve summary quality and robustness rely on training, but adversarial samples can reinforce this process improving the model's regularization and diversity of inputs and outputs [2]. However, bias remains an attack surface for this technique, although adversarial training can be used to mitigate it without compromising quality [8,19]. In the multilingual context, this problem becomes even harder, especially when working in cross-lingual contexts, as these adversarial samples transfer across languages, becoming effective in different languages, particularly in those languages of low resources or dialects. Therefore, we need new methodologies to clarify the quality of these summaries [11,21]. Finally, different evaluation metrics chosen for evaluating summary quality, such as BLEU or ROUGE, manipulate the target of summarization because it depends on metric quality and how well they understand the text. Consequently, current methods employ embeddings or LLMs to improve semantic equivalence measurement between summaries and reference summaries [3,21].

Based on the limitations and problems identified in the literature, different communities have begun working on extracting facts from original texts, employing fact comparison with summaries to enable reference-free evaluation (i.e., no human summaries) [5]. This approach accommodates multiple summaries. However, current techniques are also limited. The way we process the original text can lead to non-deterministic outcomes, where different summaries are generated. Moreover, variations in text production locations and linguistic characteristics can introduce idiosyncratic components that affect summary quality. For instance, regional languages and dialects in English or Spanish often exhibit distinct features that can alter fact extraction accuracy under complex embeddings.

Our paper aims to measure how idiosyncratic components affect the summary quality. For this purpose, our system will create an adversarial attack that rephrases the original text by replacing specific words with synonyms in order to manipulate summarization quality metrics. The goal is to determine which summarization metric is the most robust such attacks. This is crucial because it

enables us to select the most suitable technique for evaluating summary quality under regional conditions in a multi-lingual context. We propose a new methodology, named MuSEvo, based on evolutionary computation, that generates this rephrasing. MuSEvo aims to preserve the semantics of the original text while replacing words with synonyms to modify summaries. It will then evaluate and compare the quality of different summarization metrics in response to these idiosyncratic changes.

Our results show that traditional metrics, such as ROUGE and BLEU, are vulnerable to these adversarial attacks, while metrics based on large language models (LLMs) demonstrate greater robustness. We evaluated our system employing a multi-lingual scenario in three different languages, concretely, English, Spanish and French. To guide our investigation and assess the impact of these adversarial manipulations, we address the following research questions:

RQ1: *What is the most robust summarization evaluation metric to MuSEvo's attack across different languages?*
RQ2: *How do the dimensions and complexity of the search space vary across different languages?*
RQ3: *What is the distribution of the attack's effectiveness, measured by hypervolume, in relation to text complexity across languages?*

The contributions of this paper are as follows:

- We present an adversarial attack framework for summarization systems that considers both the source text and the summary as part of the search space.
- We design a multi-objective evolutionary algorithm that reduces metric scores by altering idiosyncratic word choices while preserving semantic similarity.
- We provide an analysis identifying which metrics are most robust to manipulations in the source text and summaries, and propose LLM-based metrics as a promising alternative.

2 Methodology

The proposed system aims to generate adversarial attacks targeting both a summarization model and its evaluation process, with the core objective of perturbing the original input text through controlled syntactic modifications while maintaining its semantic meaning. These perturbations serve to assess the robustness of the summarization system. Figure 1 shows MuSEvo's workflow. It begins with an input, comprising the original document and its reference summary (generated using the MT5SUM model [9]). Subsequently, `PyMultiDictionary` [15] generates candidates for synonyms that form the search space. This framework is compatible with English, Spanish, and French, our target languages. The selection of these synonyms becomes the search space for the evolutionary algorithm, to which MuSEvo will apply its aim: finding adversary variations.

A Multi-Objective Evolutionary Algorithm (MOEA), implemented in `pymoo` [1], performs the search process. The search checks various synonyms that replace the original words, and subsequently, it performs grammatical validations to ensure linguistic correctness. Notably, many grammar checkers are focused on English, so non-English models have been addressed using a translation system, specifically the Neural Machine Translation (NMT) model [18], which maintains semantic accuracy for French and Spanish. Once the adversarial text is ready, we summarise it and measure the variance between the original summary from the model and this one, as can be seen in the final step of MuSEvo. The following sections provide a detailed explanation of each step.

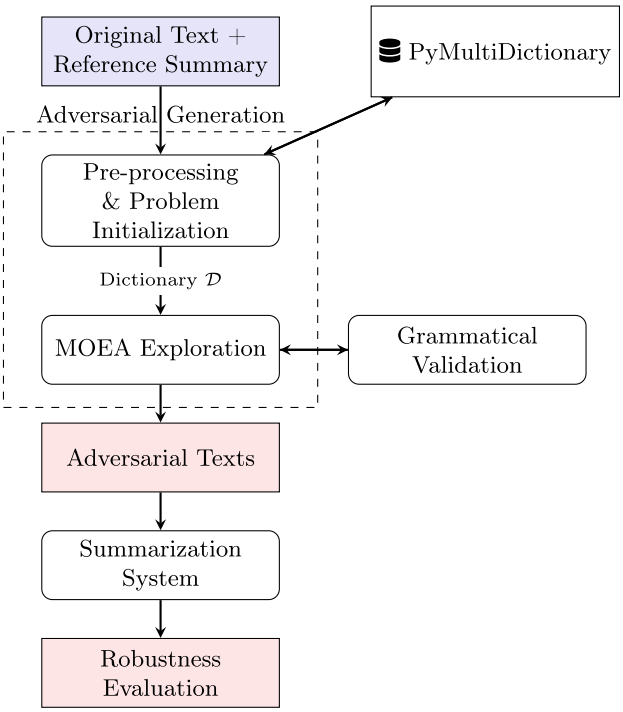


Fig. 1. Adversarial attack pipeline for summarization robustness evaluation.

2.1 Pre-processing and Problem Initialization

MuSEvo starts by tokenizing the original text and generating a dictionary of synonyms for each word, mainly focused on nouns, verbs, adjectives, and adverbs. This dictionary is generated by `PyMultiDictionary`. The dictionary defines the search space that will be optimized through the MOEA. The words are sorted by similarity in this dictionary, where the first ones are the most similar. The

dictionary is then indexed, so every word that has been identified as a potential synonym for replacement has a number of indexed candidates to substitute the original token. This is bounded using the following equation:

$$x_i \in [0, |\mathcal{S}(w_i)|] \cap \mathbb{Z} \tag{1}$$

where w_i represents the i-th token (or gene) and $\mathcal{S}(w_i)$ denotes its synonym set, ensuring valid substitutions through the bounds. Therefore, each individual $\mathbf{x}$ in the population is encoded as a vector:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{X}_1 \times \mathbb{X}_2 \times \cdots \times \mathbb{X}_n \subset \mathbb{Z}^n \tag{2}$$

where n represents the total number of tokens in the input text. Consider the phrase *"a blue house"* as an illustrative case:

- *"a"* (no synonyms) $\rightarrow x_1 \in [0, 0]$ (fixed)
- *"blue"* $\rightarrow x_2 \in [0, 2]$ ($\mathcal{S}$("azure", "cobalt"))
- *"house"* $\rightarrow x_3 \in [0, 3]$ ($\mathcal{S}$("home," "residence", "dwelling"))

For instance, let's consider an example with $\mathbf{x} = [0, 0, 2]$. This would generate a perturbed phrase, such as *"a blue residence"*, by replacing the third token. Tokens that have no synonyms remain fixed, thereby maintaining the integrity of the original content.

2.2 MOEA Exploration

MuSEvo's search process begins with an initial population. This population encodes the original text as an individual, with all alleles starting at zero. To generate new candidates, controlled randomization is employed, assigning each potential replacement of a token a specific probability (configured through a parameter). This yields a particular list of replacements, as demonstrated in Fig. 2.

This approach guarantees that initial variations remain linguistically valid and semantically proximate to the source text, as variations are restricted to: (1) tokens with available synonyms, (2) substitutions within the predefined lexical bounds, and (3) probabilistic modification rates that prevent excessive divergence. Subsequent genetic operations then evolve these initial candidates toward adversarial examples that challenge summarization systems while preserving grammatical correctness.

Base:	0	**0**	**0**	0	**0**	0	**0**
Generated:	0	**2**	**1**	0	**3**	0	**1**

Fig. 2. Base individual (all-zero encoding) and generated variant for initial population. Bold values show synonym indices from the substitution dictionary.

Given the integer-array codification of the problem, the proposed crossover operators that have been selected for the evolutionary process are Two-Point Crossover (TPX) and Uniform Crossover (UX), which work for combinatorial problems. Nevertheless, given that the dictionaries are sorted according to the similarity of the synonyms to the original term, we also tested an integer version of Simulated Binary Crossover (SBX), where the continuous values are rounded to integers. This operator is illustrated in Fig. 3. Additionally, a domain-specific mutation operator is applied to effectively explore the search space, as also shown in the figure. This mutation randomly modifies token indices within their predefined synonym bounds defined in Eq. (1), where the mutation probability (p_{mut}) controls exploration intensity while preserving semantic validity:

$$x_i' = \begin{cases} \text{randint}(0, x_i) & \text{if } r < p_{\text{mut}} \\ x_i & \text{otherwise} \end{cases} \tag{3}$$

where $r \sim U(0, 1)$. The operator enforces that all mutations are integer-only and bounded by language-specific synonym dictionaries ($\mathbf{w_i}$). These ensure discrete, valid token substitutions throughout the evolutionary process.

Parent 1:	1	2	3	0	2	0	3	1
Parent 2:	1	1	4	2	2	3	8	4
Offspring 1:	1	1	3	1	2	1	3	2
Offspring 2:	1	2	4	1	2	2	6	3

Fig. 3. Genetic operators example: where genes inside boxes indicate crossover-applied positions (SBX), and where genes inside oval boxes indicate the mutations positions (8 became 6).

The fitness evaluation begins by decoding each individual into a textual representation, followed by the application of grammar correction tailored to the target language. For English, the `language tool python` library [13] is employed directly for this purpose. Unfortunately, this library was not effective enough for other languages, and up to our knowledge, there was no other specific library to do this task proficiently. So for non-English languages, we utilize pretrained OPUS-MT neural machine translation models [20] to translate the text into English, apply grammatical corrections, and subsequently translate it back to the original language. The corrections are performed using the OPUS-MT's translation system, which maintains semantic accuracy and improves grammatical accuracy across languages.

The fitness function evaluates two distinct objectives.

1. The first objective is to *maximise the semantic similarity between the original text and the new one generated by each individual.* This is measured using

the `MiniLM-L12-v2` [16], which creates vectors from a sentence embedding optimised for paraphrase detection. We apply cosine distance to measure their similarity.
2. The second objective focuses on the summary, aiming to *minimise the similarity between the original representations and the newly generated summaries.* The model used to create the new summary is the MT5SUM model [9], and both summaries share the MiniLM embedding space, allowing for cosine similarity to be applied across them.

To select the most suitable algorithm for multi-objective optimization, we have identified three state-of-the-art MOEAs spanning three principal paradigms: `Non-dominated Sorting Genetic Algorithm II (NSGA-II)`; `Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D)`; and `S Metric Selection EMOA (SMS-EMOA)`. All of these algorithms have been implemented using `pymoo` [1], and they are compared in Sect. 3.2. In the experimental section, we evaluate the performance of each strategy under identical adversarial generation settings to empirically determine the most effective optimization approach for our multilingual perturbation task. As the true Pareto set is unknown for this problem, the Hypervolume metric is employed for the evaluation of the effectiveness of the solutions. This metric allows the comparison of the three proposed multi-objective approaches in terms of convergence and diversity of the approximated Pareto set.

The complete search procedure for generating adversarial text variations is formalized in Algorithm 1, which brings together the initialization, variation, evaluation, and selection steps outlined above. As can be seen in Line 23, f_1 is assigned a negative value because similarity is to be maximized, and pymoo framework is based on minimization. The algorithm evolves the population iteratively and terminates either when the maximum number of generations ($G_{\max}$) is reached, or when improvements in the objective space become negligibly small across consecutive generations, as determined by the `RobustTermination` method from the `pymoo` framework.

2.3 Robustness Evaluation

We will provide an evaluation employing both classical metrics and LLM-based ones. For that, we will employ MuSE, which contains different evaluation metrics extracted from the state of the art. Concretely, we will employ ROUGE and BLEU as the classical metrics [6], and we will employ a factuality strategy using the large language models. This latter strategy will measure the average across: factuality, completeness, and conciseness.

Classical Metrics. For the evaluation, we employ two classical metrics extensively used in the literature: BLEU and ROUGE [6].

ROUGE measures n-gram overlapping recall. For each reference text S, extract all n-grams. For each n-gram ($gram_n$) in the reference(s), count how

Algorithm 1. Adversarial Text Variation Search via MOEA

Require: Original text T, reference summary S_{ref}, synonym dictionary $\mathcal{D}$, population size pop_size, maximum number of generations G_{max}, initialization rate p_{init}, mutation rate p_{mut}

Ensure: Set of adversarial texts $\mathcal{A}$

Initial Population:
Encode T as all-zero vector $\mathbf{v}_0$ ▷ Base genotype
for $i \leftarrow 1$ to pop_size **do**
  Initialize $\mathbf{v}_i \leftarrow \mathbf{v}_0$
  for each token position j in T **do**
    if rand() $< p_{\text{init}}$ and T_j has synonyms in $\mathcal{D}$ **then**
      $\mathbf{v}_i[j] \leftarrow$ random_index($\mathcal{D}[T_j]$) ▷ Controlled randomization
  $I_i \leftarrow \mathbf{v}_i$
  Add I_i to P_0
Evolution:
$g \leftarrow 1$
while $g <= G_{\max} \vee \neg$RobustTermination() **do**
  Crossover: Apply TPX/UX/SBX crossover on parent pairs to produce offspring Q_g
  Mutation:
  for each individual $I \in Q_g$ **do**
    for each gene j in I **do**
      if rand() $< p_{\text{mut}}$ **then**
        $I[j] \leftarrow$ randint(0, $|\mathcal{D}[T_j]|$) ▷ Bounded by synonym dictionary
  Fitness Evaluation:
  for each individual $I_k \in P_g \cup Q_g$ **do**
    $T_k \leftarrow$ decode($I_k, \mathcal{D}$) ▷ Reconstruct perturbed text
    $T_k \leftarrow$ correct_grammar(T_k) ▷ Ensure grammatical correctness
    $f_1 \leftarrow -$(cosine_sim(embed(T)), embed(T_k)) ▷ Semantic preservation
    $S_k \leftarrow$ mT5Sum(T_k)
    $f_2 \leftarrow$ cosine_sim(embed(S_{ref}), embed(S_k)) ▷ Adversarial divergence
    fitness(I_k) $\leftarrow [f_1, f_2]$
  Selection: Apply NSGA-II, MOEA/D or SMS-EMOA $\rightarrow P_{g+1}$
  $g \leftarrow g + 1$
$\mathcal{A} \leftarrow \{$correct_grammar(decode($I, \mathcal{D}$)) $\mid I \in P_{G_{\max}}\}$

many times it appears in both the candidate text T_c and the reference text T_r, this is $\text{Count}_{\text{match}}$. Then, ROUGE-N sums up all matches and divide by the total number of n-grams in the reference(s).

$$\text{ROUGE-N} = \frac{\sum_{\text{S} \in \text{References}} \sum_{\text{gram}_n \in \text{S}} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{S} \in \text{References}} \sum_{\text{gram}_n \in \text{S}} \text{Count}(\text{gram}_n)} \tag{4}$$

During the evaluation, we employ **ROUGE-L** which takes into account the longest common subsequence (LCS). It is defined as:

$$\text{ROUGE-L} = \frac{LCS(T_c, T_r)}{\text{length of reference}} \tag{5}$$

BLEU evaluates n-gram precision. It measures the overlap between n-grams of T_c and T_r:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{6}$$

in the equation BP is the brevity penalty, w_n is the n-grams weights, normally, $w_n = \frac{1}{N}$ p_n is the n-gram precision. BP can be calculated as:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{7}$$

where r is the reference length and c is the candidate length.

LLM-Based Metrics. Newer techniques apply large language models (LLMs) to assess summaries on dimensions such as coherence, factuality, completeness, and information density [10]. These metrics, named fact-based metrics, use LLMs to extract and compare facts from the different texts. Let T_1 and T_2 be two texts. By employing an LLM, the metric extracts the sets of facts F_{T_1} and F_{T_2}, corresponding to the respective facts of each text. Let define P as all possible pairwise combinations of facts between F_{T_1} and F_{T_2}. Each pair $p \in P$ is simply two facts from each text. Considering that there are facts in the text that might not be representative enough with respect to the facts of the compared text, the metric performs a filter, keeping only the pair with the maximum similarity from both sets, assuming that the similarity is above a specific threshold τ. For example, choosing a single $f^i_{T_1} \in F_{T_1}$, and every $f^j_{T_2} \in F_{T_2}$, P contains every possible pair, however, the metric keeps only the pair with maximum similarity $p^* = (f^*_{T_1}, f^*_{T2})$ to improve the quality of factuality pairing, assuming that $sim(p^*) \geq \tau$, otherwise we set $sim(p^*) := 0$. This set is defined as P^*. Then, Factuality is the average similarity across all possible pairs in P^*, therefore:

$$\text{Fact}(T_1, T_2) = \frac{1}{|P^*|} \sum_{p \in P^*} sim_{Fact}(p) \tag{8}$$

where

$$sim_{Fact}(p) = \begin{cases} 0 & \text{if } sim(p) < \tau \\ sim(p) & \text{otherwise} \end{cases} \tag{9}$$

Completeness focuses on counting the number of facts from T_1 that have proper correspondence with respect to the facts in T_2. Similarly to factuality it is based on P^* and counts the non-empty pairs.

$$\text{Comp}(T_1, T_2) = \frac{1}{|P^*|} \sum_{p \in P^*} sim_{Comp}(p) \tag{10}$$

$$sim_{Comp}(p) = \begin{cases} 0 \text{ if } sim(p) < \tau \\ 1 \text{ otherwise} \end{cases} \tag{11}$$

Finally, Density balances both metrics and it is defined as:

$$\text{Den}(T_1, T_2) = \frac{1}{2}\left(\text{Comp}(T_1, T_2) + \text{Fact}(T_1, T_2)\right) \tag{12}$$

All metrics values are between the $[0, 1]$ range, where 0 represents the minimum value and 1 the maximum.

3 Experimental Setup

The experimental evaluation is designed to provide empirical evidence for the research questions previously defined. We focus on assessing the impact of MuSEvo's syntactic manipulations on multilingual summarization. We start by presenting the dataset, followed by an explanation of how the parameters were tuned, and conclude with a comparative analysis of the algorithms employed in our experiments.

3.1 Dataset

The dataset used is the XL-Sum dataset [9], which comprises several news texts in 44 languages, thereby making it a multilingual dataset that includes both high- and low-resource languages. It contains 1.35 million articles and human summaries as references from the BBC. This dataset has been extensively utilized in the literature due to its annotations, and its cross-lingua nature. In our experiments, we selected three languages from the dataset: English, French, and Spanish. For each of these languages, we selected 10 articles, sorted by computational complexity of the synonym search space generated by MuSEvo. This enables us to control the evaluation while maintaining a diverse balance between languages and the impact of the algorithms on various quality metrics.

3.2 Parameter Selection and Algorithm Performance Comparison

To systematically determine optimal parameter values, we conducted preliminary experiments evaluating classical algorithmic parameters, such as the population size or the mutation and crossover probabilities, using Bayesian Optimization with HyperBand (BOHB) [7]. For this initial tuning phase, we selected one of the English texts with the lowest dictionary-based complexity (ID 62) from our stratified dataset. The results of the best-performing configuration identified are summarized in Table 1. In addition, for MOEA/D the Tschebycheff scalarization function and *Das and Denis* method for reference directions computation were used.

Once the general parameters were fixed, we executed each evolutionary algorithm (`NSGA-II`, `SMS-EMOA`, `MOEA/D`) 31 times in order to evaluate both performance and robustness. This analysis supports the selection of the most effective and reliable algorithm for generating adversarial text variations.

Table 1. Best-performing parameter configuration obtained.

Parameter	Value
Population size (pop_{size})	50
Initialization probability (p_{init})	0.7
Crossover type	*SBX*
Crossover probability ($p_{\text{crossover}}$)	0.7
Crossover distribution index (η)	10
Mutation probability (p_{mut})	0.2

To assess the robustness of the algorithms, we use statistical measures derived from the normalized hypervolume results, including the mean, standard deviation, and the coefficient of variation (CV). The CV is calculated as the ratio between the standard deviation (σ) and the mean (μ) ($CV = \frac{\sigma}{\mu}$), which provides a normalized measure of relative variability.

We employ the coefficient of variation (CV) to measure robustness levels, defining three distinct thresholds. When the CV is below 0.05, we consider *High* robustness; when it falls between 0.05 and 0.1, we regard it as *Acceptable*; and for values exceeding or greater than 0.1, we classify it as *Low* robustness. The objective of this coefficient is to identify which algorithm demonstrates more consistency and stability across multiple executions.

Table 2. Comparative performance and robustness metrics based on Hypervolume for each algorithm.

Algorithm	Mean	Std. Dev.	CV (σ/μ)	Robus. Level
SMS-EMOA	0.8253	0.0462	0.0560	Acceptable
NSGA-II	0.7971	0.0716	0.0898	Acceptable
MOEA/D	0.6061	0.1149	0.1896	Low

Table 2 indicates that SMS-EMOA outperforms all other algorithms in terms of effectiveness and robustness, achieving a higher hypervolume, superior optimization performance, and greater consistency across executions, primarily due to lower variability. In contrast, MOEA/D exhibits poorer performance, becoming the weakest algorithm and displaying the highest variability, rendering it less reliable; while NSGA-II strikes a balance between performance and robustness.

We have also applied the Friedman test to evaluate whether observed differences are statistically significant, including the Nemenyi post-hoc analysis [4]. The test reveals highly significant results ($\chi^2 = 43.42$, $p = 3.73 \times 10^{-10} < 0.0001$), confirming that algorithms exhibit statistically distinct performance and yield different outcomes. The Nemenyi post-hoc test results (Table 3) indicate that MOEA/D performs significantly worse than both NSGA-II ($p = 0.001$)

and SMS-EMOA ($p = 0.001$). However, no statistically significant difference was observed between NSGA-II and SMS-EMOA ($p = 0.2247$), despite the latter showing superior average performance and greater robustness.

Table 3. Post-hoc Nemenyi test p-values between algorithms.

	MOEA/D	NSGA-II	SMS-EMOA
MOEA/D	–	0.001	0.001
NSGA-II	0.001	–	0.22467
SMS-EMOA	0.001	0.22467	–

In summary, SMS-EMOA not only achieved the highest mean performance but also exhibited the greatest robustness across runs. These findings, supported by both descriptive and inferential analyses, justify the selection of SMS-EMOA as the most appropriate algorithm.

Finally, a comparative analysis of crossover operators (TPX, UX and SBX) was conducted using the same robustness and statistical framework based on the hypervolume indicator. SBX demonstrated the highest robustness, exhibiting the lowest coefficient of variation (CV = 0.056). This better performance demonstrates that the order of dictionaries according to similarity to the original term can in fact affect the exploitation in evolutionary algorithms. Although the Friedman test did not indicate statistically significant differences among operators ($\chi^2 = 4.064$, $p = 0.131$), SBX is selected as the crossover operator for subsequent experiments due to its more consistent empirical performance.

4 Results

This section presents a scalability study designed to evaluate the performance of the proposed evolutionary approach across varying levels of textual complexity and different languages. The following subsections provide the empirical results and analysis required to answer RQ1 through RQ3, based on the experimental design established in Sect. 3.

Following the configuration in Sect. 3.2, 10 input texts per language were selected and ordered by increasing complexity based on their search space size. We repeat the experiments ten times to account for common variations in evolutionary algorithms, and conduct an analysis of the complexity of the search space and linguistic aspects.

We use a Ubuntu 22.04 machine with an Intel Core i7-11800H processor (2.30 GHz), 32 GB of RAM, and an NVIDIA GeForce RTX 3070 GPU. The dataset and evolutionary algorithm are publicly available on a GitHub repository[1], which also includes Jupyter Notebooks designed to replicate all results and

[1] The final link to the repository will be provided upon publication.

visualizations. Furthermore, we have made our code and data available under the GNU General Public License version 3. Additionally, the authors are available for any further technical inquiries.

4.1 RQ1. Summarization Quality Metrics

Research question 1 focuses on the robustness against adversarial attacks. We consider that a system is more stable when its quality metric score after an attack remains higher, indicating greater resilience.

During our evaluation, we focus on two classical metrics (BLEU, ROUGE) and five LLMs using factuality-based metrics. Our analysis is shown in Table 4. The results show that BLEU becomes the least reliable metric across all languages due to its requirement for exact overlapping of n-grams and penalization of paraphrasing, making it less suitable for abstract summarization compared to LLMs. On the other hand, TinyLlama reveals limitations in extracting facts from adversarial perturbations. The best models are Mistral and Gemma3, which exhibit robustness, particularly Mistral achieving good results in Spanish, while Gemma3 demonstrates consistency across all languages.

Notably, Llama 3.2 appears more sensitive to Spanish data than to English. Regarding the Pareto front, no significant differences were observed between best, median, or worst solutions, suggesting that metric stability is relatively consistent regardless of the final optimization level. English and Spanish tend to yield the best results, whereas French shows weaker performance due to its higher structural and conjugation complexity.

RQ1 Answer. Mistral and Gemma3 are the most robust metrics against MuSEvo's attack. Mistral shows the highest stability in Spanish, while Gemma3 is more consistent across languages. BLEU and TinyLlama are the least reliable due to structural rigidity and limited semantic capacity.

4.2 RQ2. Search Space and Linguistic Complexity

This research question analyzes the search space dimensions and linguistic variability across languages. As shown in Table 5, French exhibits the largest and most heterogeneous synonym coverage (up to 6,292 total synonyms), leading to irregular search behavior. English and Spanish display more balanced distributions, contributing to steadier performance.

To assess internal variations, we applied a Kruskal–Wallis test followed by Dunn's post-hoc analysis with Holm correction. Significant differences were found in all languages ($p < 0.001$). In English, text 59 showed broad differences; in Spanish, texts 89, 75, and 4 stood out; while in French, differences were marked between extreme complexity levels (e.g., 22 vs 93/53). Despite these variances, a cross-language Kruskal-Wallis test revealed no significant differences in normalized hypervolume ($H = 3.145, p = 0.208$), suggesting that complexity distributions are statistically comparable across languages overall.

Table 4. Summarization quality metrics grouped by metric type and language.

	English			Spanish		
Metric	Best	Median	Worst	Best	Median	Worst
BLEU	.134±.22	.138±.24	.098±.20	.116±.19	.148±.16	.189±.24
ROUGE	.320±.21	.328±.24	.296±.18	.307±.18	.326±.16	.343±.24
DeepSeek	.260±.36	.345±.38	.300±.36	*.320±.36*	.416±.32	.439±.35
Gemma3	*.333±.34*	**.376±.36**	*.383±.32*	.314±.36	*.422±.35*	.398±.37
Llama3.2	.218±.34	.310±.39	.235±.35	.289±.37	.353±.37	*.448±.40*
Mistral	**.379±.38**	*.357±.38*	**.384±.37**	**.515±.36**	**.617±.29**	**.669±.29**
Tinyllama	.165±.28	.200±.29	.176±.30	.126±.28	.245±.33	.240±.34

	French		
Metric	Best	Median	Worst
BLEU	.052±.10	.067±.12	.051±.10
ROUGE	**.226±.15**	**.236±.16**	**.232±.14**
DeepSeek	.175±.30	.181±.28	*.214±.33*
Gemma3	*.187±.31*	*.218±.34*	.198±.32
Llama3.2	.107±.26	.126±.28	.086±.23
Mistral	.167±.31	.172±.30	.187±.30
Tinyllama	.099±.25	.119±.28	.074±.21

RQ2 Answer. Search space dimensions and linguistic complexity vary significantly within each language, affecting algorithmic stability. However, the overall hypervolume distributions are statistically comparable across English, Spanish, and French.

4.3 RQ3. Evolutionary Performance and Hypervolume

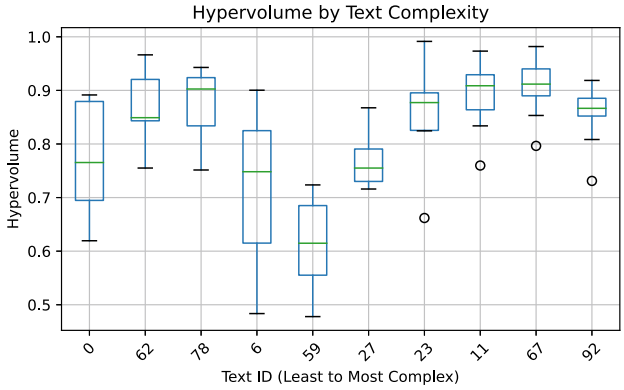
The effectiveness of MuSEvo is captured by the hypervolume (HV) of the Pareto front. As illustrated in Fig. 4, HV tends to increase with text complexity across all languages. Higher synonym availability provides the algorithm with a richer combinatorial space, enabling more effective exploration of the trade-off between semantic preservation and metric degradation.

Table 5. Search space characteristics by text and language, reorganized for compactness.

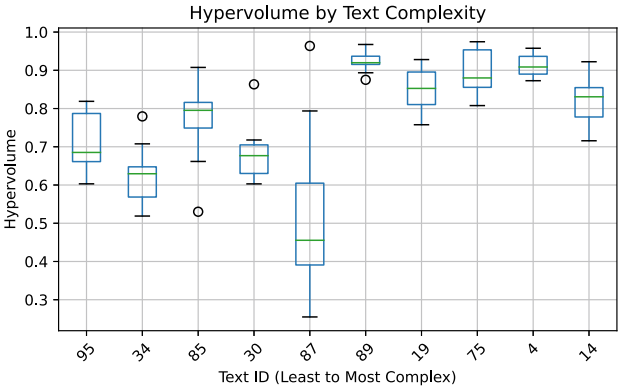
Lang.	Txt	Tot. Syn.	Avg. Syn.	Txt	Tot. Syn.	Avg. Syn.
English	0	131	5.46 ± 5.80	27	1731	6.74 ± 6.14
	62	148	6.73 ± 4.10	23	1833	8.04 ± 6.15
	78	177	5.36 ± 6.13	11	2101	6.96 ± 6.42
	6	300	6.52 ± 7.01	67	2189	6.51 ± 6.13
	59	806	7.01 ± 6.19	92	3797	8.48 ± 7.50
Spanish	95	120	4.80 ± 3.52	89	1479	5.87 ± 5.01
	34	148	5.92 ± 4.95	19	1880	6.44 ± 5.57
	85	161	5.96 ± 4.00	75	2106	6.32 ± 5.18
	30	233	7.28 ± 5.87	4	2551	6.36 ± 5.03
	87	324	6.23 ± 5.33	14	3105	6.63 ± 5.59
French	22	194	9.24 ± 9.10	93	577	13.11 ± 11.42
	21	200	11.11 ± 7.85	28	838	14.45 ± 15.10
	45	322	9.76 ± 9.84	25	3255	12.62 ± 14.73
	49	358	14.92 ± 17.20	53	3792	14.20 ± 14.67
	61	450	10.98 ± 12.70	72	6292	14.14 ± 15.04

In English and Spanish, HV improves consistently with complexity. In French, while the trend holds for low levels, extreme complexity (e.g., texts with > 6000 synonyms) introduces larger variation in HV. This indicates that as the search space expands exponentially, optimization faces challenges due to combinatorial explosion. Nevertheless, the algorithm effectively leverages linguistic richness to identify adversarial examples while maintaining semantic coherence.

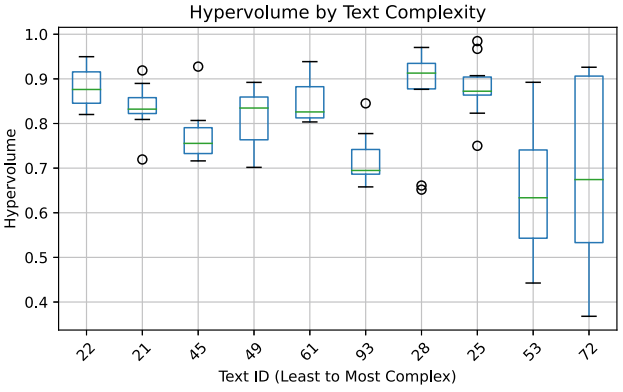
RQ3 Answer. Hypervolume tends to increase with text complexity across all languages. The evolutionary algorithm leverages richer synonym spaces to achieve more effective optimization and exploration.



(a) English



(b) Spanish



(c) French

Fig. 4. Hypervolume distribution by text complexity across languages

5 Conclusions

MuSEvo demonstrates how adversarial changes based on idiosyncratic components significantly disrupt the quality of summarization metrics. This highlights the limitations of these metrics, making them less reliable, as they can be easily degraded by small text permutations that do not compromise the original text's semantics. The metrics are especially sensitive to multiple languages, even those where new methods relying on factual checks may struggle.

Our work concludes that, depending on how the metric is designed, whether based on n-grams, such as BLEU or ROUGE, or LLMs, like Mistral and Gemma 3, its robustness varies significantly (RQ1). The LLM metrics prove to be more resilient. The main reason for this sensitivity lies in the complexity of the search space of synonyms. A larger search space allows for a wider range of changes, making it increasingly challenging for evaluation metrics to remain robust (RQ2). Consequently, the complexity of the search space has a direct impact on quality, as there are more lexical resources available to be exploited. For instance, manipulations can be particularly disruptive in French due to the complex nature of this language. Furthermore, our evolutionary algorithm demonstrates its ability to scale with varying text complexities (RQ3), enabling it to launch more sophisticated adversarial attacks and evaluate advanced summarization systems.

For future work, we aim to improve the quality of metrics that utilize factuality, particularly those that can withstand adversarial changes, and develop methods to rephrase or restructure grammatical changes as well. Furthermore, we plan to extend our approach to other languages with distinct properties, such as Chinese or Arabic, and incorporate enhanced grammatical correction techniques during the evolutionary process to create an even more complex search space for adversarial attacks.

Acknowledgments. The authors thank the UKRI Trustworthy Autonomous Systems Hub (EP/V00784X/1 and EP/V026801/2), and the Comunidad Autónoma de Madrid under the ALENTAR-J-CM project (TEC-2024/COM-224) for their support.

Competing Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Blank, J., Deb, K.: pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020). https://doi.org/10.1109/ACCESS.2020.2990567
2. Cao, B., Guan, X., Bao, S., Wu, J., Fan, J.: Optimizing representation for abstractive multi-document summarization based on adversarial learning strategy. IEEE Trans. Cogn. Dev. Syst. (2025). https://doi.org/10.1109/TCDS.2025.3563357
3. Chen, Y., Eger, S.: Menli: robust evaluation metrics from natural language inference. Trans. Assoc. Comput. Linguist. **11**, 804–825 (2023). https://doi.org/10.1162/tacl_a_00576
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006). https://doi.org/10.5555/1248547.1248548

5. Dierickx, L., Van Dalen, A., Opdahl, A.L., Lindén, C.G.: Striking the balance in using LLMs for fact-checking: a narrative literature review. In: Multidisciplinary International Symposium on Disinformation in Open Online Media, pp. 1–15. Springer, Heidelberg (2024). https://doi.org/10.1007/978-3-031-71210-4_1
6. Ermakova, L., Cossu, J.V., Mothe, J.: A survey on evaluation of summarization methods. Inf. Process. Manag. **56**(5), 1794–1814 (2019). https://doi.org/10.1016/j.ipm.2019.04.001
7. Falkner, S., Klein, A., Hutter, F.: Bohb: robust and efficient hyperparameter optimization at scale. In: Proceedings of the 35th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 80, pp. 1437–1446. PMLR (2018). https://doi.org/10.48550/arXiv.1807.01774
8. Gupta, M., Varimalla, N.R., Deas, N., Subbiah, M., McKeown, K.: Advsumm: adversarial training for bias mitigation in text summarization. arXiv preprint arXiv:2506.06273 (2025). https://doi.org/10.48550/arXiv.2506.06273
9. Hasan, T., et al.: XL-sum: large-scale multilingual abstractive summarization for 44 languages. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 4693–4703. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.findings-acl.413
10. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-eval: NLG evaluation using gpt-4 with better human alignment. arXiv preprint arXiv:2303.16634 (2023). https://doi.org/10.48550/arXiv.2303.16634
11. Mathur, V., Dadu, T., Aggarwal, S.: Evaluating neural networks' ability to generalize against adversarial attacks in cross-lingual settings. Appl. Sci. **14**(13), 5440 (2024). https://doi.org/10.3390/app14135440
12. Menéndez, H.D., Dakhama, A.: Automatic summarization evaluation: methods and practices. In: IFIP International Conference on Testing Software and Systems, pp. 169–181. Springer, Heidelberg (2024). https://doi.org/10.1007/978-3-031-80889-0_12
13. Morris, J.: Python wrapper for languagetool (2021). https://pypi.org/project/language-tool-python/, python wrapper for LanguageTool
14. Qazvinian, V., et al.: Generating extractive summaries of scientific paradigms. J. Artif. Intell. Res. **46**, 165–201 (2013). https://doi.org/10.1613/jair.3732
15. Pablo Pizarro R.: Pymultidictionary: a multi-language dictionary module (2021). https://github.com/ppizarror/PyMultiDictionary. versión 1.3.1
16. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1410
17. Shi, T., Keneshloo, Y., Ramakrishnan, N., Reddy, C.K.: Neural abstractive text summarization with sequence-to-sequence models. ACM Trans. Data Sci. **2**(1), 1–37 (2021). https://doi.org/10.1145/3419106
18. Stahlberg, F.: Neural machine translation: a review. J. Artif. Intell. Res. **69**, 343–418 (2020). https://doi.org/10.1613/jair.1.12007
19. Thota, P., Nilizadeh, S.: Attacks against abstractive text summarization models through lead bias and influence functions. In: Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 13727–13741. Association for Computational Linguistics (2024). https://doi.org/10.18653/v1/2024.findings-emnlp.804
20. Tiedemann, J., et al.: Democratizing neural machine translation with OPUS-MT. Lang. Res. Eval. **58**, 713–755 (2023). https://doi.org/10.1007/s10579-023-09704-w

21. Zhang, R., Ouni, J., Eger, S.: Cross-lingual cross-temporal summarization: dataset, models, evaluation. Comput. Linguist. **50**(3), 1001–1047 (2024). https://doi.org/10.1162/coli_a_00519
22. Zheng, J., Hu, T., Cong, T., He, X.: Cl-attack: textual backdoor attacks via cross-lingual triggers. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 26427–26435 (2025). https://doi.org/10.1609/aaai.v39i25.34842