# Evolutionary Optimization of Instruction-Tuned Large Language Model Inference Parameters for Spanish Text Simplification

Nelson Pérez-Rojas
Doctorado en Ciencias Naturales para el Desarrollo (DOCINADE), Instituto Tecnológico de Costa Rica, Universidad Nacional, Universidad Estatal a Distancia
San José, Costa Rica
nperez@itcr.ac.cr

Danny Xie-Li
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
dxie@ic-itcr.ac.cr

Martín Solís
Instituto Tecnológico de Costa Rica
San José, Costa Rica
marsolis@itcr.ac.cr

Saúl Calderón-Ramírez
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
sacalderon@itcr.ac.cr

Sebastián Rojas-Gónzalez
Ghent University - IMEC
Gante, Belgium
sebastian.rojasgonzalez@imec.be

Víctor Ortíz-Ruíz
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
voruiz@gmail.com

## Abstract

Large Language Models (LLM) have become valuable tools for automatic text simplification, yet their output quality is highly sensitive to inference-time parameters such as temperature, repetition penalty, top-p, and top-k. These parameters are typically set heuristically rather than systematically optimized. In this work, we apply Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to identify high-performing inference configurations for an instruction-tuned Large Language Model (LLM) fine-tuned for Spanish Text Simplification (TS) on the Financial Education Corpus IN SpAnish (FEINA) benchmark. Using System output Against References and against the Input sentence (SARI) as the optimization objective and Sentence Bidirectional Encoder Representations from Transformers (SBERT) as a monitoring metric, CMA-ES converges in 9 of 25 generations, improving test-set SARI from 36.57 (default) to 38.87 (+6.3%) while increasing SBERT from 0.85 to 0.88. Grid search analysis reveals that repetition penalty and temperature are the dominant factors influencing simplification quality, while top-$k$ and maximum tokens have negligible effects. The strong positive correlation between SARI and SBERT across all 2,500 evaluated configurations provides empirical support for the single-objective approach within the studied setting. Our framework offers a replicable methodology for inference parameter optimization applicable to Spanish and other under-resourced languages in Latin American Natural Language Processing (NLP) contexts.

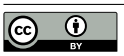
## CCS Concepts

• **Software and its engineering**; • **Applied computing** → **Computers in other domains**;

## Keywords

Large Language Models, Automatic Text Simplification, Covariance Matrix Adaption Evolution Strategy, Hyperparameter Tuning, Spanish NLP, Low-Resource Languages

**ACM Reference Format:**
Nelson Pérez-Rojas, Danny Xie-Li, Martín Solís, Saúl Calderón-Ramírez, Sebastián Rojas-Gónzalez, and Víctor Ortíz-Ruíz. 2026. Evolutionary Optimization of Instruction-Tuned Large Language Model Inference Parameters for Spanish Text Simplification. In *Genetic and Evolutionary Computation Conference (GECCO Companion '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3795101.3814720



This work is licensed under a Creative Commons Attribution 4.0 International License.
*GECCO Companion '26, San Jose, Costa Rica*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2488-6/2026/07
https://doi.org/10.1145/3795101.3814720

## 1 Introduction

Inference-time decoding parameters in LLMs directly influence the stochastic generation process, yet their systematic optimization remains largely heuristic [14]. Parameters such as temperature, top-$k$, top-$p$, and repetition penalties modulate either the rescaling or the truncation of the output token distribution, altering sampling behavior without modifying model weights. While hyperparameter optimization (HPO) has been extensively studied in machine learning [17], such work primarily focuses on training-time configurations affecting learning dynamics rather than inference-time sampling behavior.

In generative tasks such as TS, evaluation is inherently multi-dimensional. Trade-offs between structural transformation quality and semantic preservation have been reported in controllable TS [15] and in optimization of LLM-generated text [27]. However, most existing approaches rely on heuristic tuning without formally characterizing the relationship between inference parameters and evaluation metrics [28].

From an optimization perspective, the presence of multiple evaluation metrics does not automatically require multi-objective optimization. When one metric serves as the primary optimization target and others function as diagnostic indicators, single-objective optimization provides a simpler and more efficient approach.

In this work, we treat inference-time parameter calibration as a black-box optimization problem. Using Spanish text simplification

over the FEINA benchmark [20], we apply CMA-ES [7] to explore the space of inference-time parameters—including temperature, top-k, top-p, repetition penalty, and maximum generation length—and identify configurations that maximize SARI. We additionally monitor SBERT to assess whether the optimized configurations preserve semantic fidelity. We evaluate the optimization trajectory under a fixed budget of model evaluations and analyze the sensitivity of both metrics to different parameter configurations.

The contributions of this work are:

(1) **Systematic exploration of inference-time parameters:** We characterize the relationship between inference-time parameters and simplification quality in Spanish, identifying which configurations most strongly influence SARI and SBERT.
(2) **Efficient optimization via CMA-ES:** We apply CMA-ES to automatically identify high-performing inference configurations under a limited evaluation budget.
(3) **Practical framework for LLM adaptation:** We provide a replicable methodology for tuning inference parameters in LLM-based text simplification systems, with direct applicability to Spanish and other under-resourced languages.

By framing inference-time parameter tuning as a structured black-box optimization problem, this study advances LLM-based text simplification from heuristic adjustment toward principled optimization methodology.

## 2 State of the Art

### 2.1 Decoding Strategies in Text Generation

Conditional text generation requires selecting tokens from learned probability distributions, but the decoding strategy determines the fundamental trade-off between quality and diversity of the generated text [28]. This trade-off is particularly critical in TS, which simultaneously demands semantic fidelity and structural transformation. Maximization-based methods such as beam search have been shown to produce degenerate outputs — repetitive, generic, or incoherent — despite the underlying model assigning high probability to grammatically correct text [8]. Nucleus sampling (top-$p$) was proposed to address this limitation by dynamically truncating the unreliable tail of the distribution and sampling from the probability mass nucleus, yielding text that better resembles human writing in open-ended generation tasks [8]. However, decoding strategy properties do not transfer reliably across tasks [28]: directed tasks such as translation and summarization favor deterministic methods, whereas open-ended tasks benefit from stochastic strategies. TS occupies an intermediate position between these two regimes, which motivates empirical exploration of the inference-parameter space rather than relying on defaults established for other tasks.

### 2.2 Inference-Time Parameter Recommendations

Traditional inference-time optimization relies on manually tuned static hyperparameters — temperature, top-$k$, top-$p$, and repetition penalties — that are fixed across all inputs and generation steps. Several influential works provide concrete parameter recommendations for specific use cases. Repetition penalty values around 1.2 have been recommended to balance truthful generation and avoid repetition [10], a guideline now incorporated into widely used NLP libraries. Recent empirical work across the Llama, Qwen, and Mistral model families identifies temperature 0.8 with top-$p$ 0.9 as optimal for complex reasoning tasks [13], while temperatures between 0 and 2 are recommended for general LLM use [4, 12]. Top-$k$ sampling is typically recommended within the range $[20, 640]$ [8]. However, these recommendations are task-agnostic and do not account for the specific demands of Spanish TS.

### 2.3 Systematic Inference-Time Optimization

Several recent works have addressed inference-time parameter optimization through systematic, dataset-specific tuning procedures. An entropy-based method that automatically optimizes temperature without labeled data has been proposed [4], identifying an entropy turning point at which token-level entropy shifts from concave to convex, with strong results across mathematical reasoning, code generation, and question-answering tasks. A complementary grid search framework across a four-dimensional parameter space — temperature, reasoning steps, planning intervals, and top-$p$ — has been shown to identify static configurations that generalize across model architectures [19], consistently improving mathematical precision with temperatures between 0.1–0.4 and top-$p$ values of 0.85–0.98. These approaches, however, focus on reasoning tasks and do not address generative tasks such as TS, nor do they apply evolutionary optimization methods to the parameter search process.

### 2.4 Text Simplification with Large Language Models

Automatic TS has undergone a fundamental transformation with the adoption of LLMs. Traditional TS systems were evaluated primarily through n-gram-based metrics such as Bilingual Evaluation Understudy Score (BLEU) [18] and SARI [29], although these metrics present significant limitations for capturing semantic adequacy, particularly when simplifications diverge lexically from references [1].

More holistic evaluation approaches have been proposed to address these limitations, combining semantic similarity with the degree of simplification relative to the source sentence [16]. In this work, SBERT is employed as a proxy for semantic preservation, following the intuition that sentence embeddings capture meaning more robustly than lexical metrics [22]. Benchmarking across 44 LLMs for TS has shown that training method matters more than model size, with smaller, adequately fine-tuned models outperforming larger models with suboptimal adaptation [11].

### 2.5 Spanish Text Simplification

Research on automatic TS has developed predominantly for English, with Spanish receiving comparatively less attention despite the language's global reach and the particular accessibility needs of Spanish-speaking populations [25]. Early work on Spanish TS was pioneered by the Simplext project, which developed the first automatic simplification system for Spanish that addressed both lexical and syntactic complexity and evaluated it against human-produced simplifications [24, 25]. This system demonstrated that rule-based

and hybrid approaches could achieve simplification quality comparable to state-of-the-art English systems, even with fewer available language resources.

Subsequent efforts have focused on building resources for Spanish lexical simplification. The ALEXSIS dataset [6] introduced the first Spanish benchmark for lexical simplification evaluation with a high number of annotated substitutions per instance, enabling systematic comparison of simplification approaches. More recently, multilingual initiatives have incorporated Spanish as a target language, with the MultiLS framework [26] providing lexical complexity prediction and simplification annotations for Spanish educational texts, with contributions from Latin American research groups. A systematic review of research-based TS tools [5] identified several Spanish tools, including Simplext, EASIER, DysWebxia, and NavegaFácil, but noted that most are currently inoperative, highlighting the need for maintained, accessible simplification resources for Spanish.

Despite these advances, the application of LLM-based approaches to Spanish TS remains underexplored relative to English. In particular, the systematic optimization of inference-time parameters for LLMs fine-tuned on Spanish simplification data has not been previously addressed. This work targets that gap using the FEINA benchmark [20], a domain-specific corpus for financial text simplification in Spanish developed with accessibility of visually impaired users in mind.

## 3 Methodology

We apply CMA-ES as a single-objective optimization algorithm to tune the inference-time decoding parameters of Llama-3.2-1B-Instruct, a 1-billion-parameter language model fine-tuned for Spanish TS on the FEINA dataset. The model is loaded in 4-bit quantization using the Unsloth framework to enable efficient inference on a single GPU. The optimization objective is to maximize mean SARI over a randomly sampled optimization subset of 100 instances from the test set. The best configuration is subsequently evaluated on a separate held-out subset of 124 instances with no overlap with the optimization subset. The parameters subject to optimization are:

- **Top-$k$:** Limits sampling to the $k$ most probable next tokens.
- **Top-$p$:** Nucleus sampling; selects tokens whose cumulative probability mass reaches $p$.
- **Temperature:** Scales the output distribution; lower values favor more deterministic generation.
- **Repetition penalty:** Reduces the likelihood of repeating previously generated tokens.
- **Maximum new tokens:** Sets the upper bound on generated sequence length.

Empirical analysis of the evaluated configurations confirms that these parameters significantly affect SARI, with repetition penalty and temperature emerging as the dominant factors (see Section 5).

CMA-ES was configured with a fixed population size of 100, an initial step size $\sigma_0 = 0.3$, and 25 generations, yielding a total budget of 2,500 candidate evaluations.

## 4 Experimental Design

The following inference parameters are included in the optimization process, with their corresponding search ranges:

- **Top-$k$:** A discrete parameter recommended to be set between 20 and 640 [8]. Higher values increase output diversity. This parameter is frequently neglected during inference tuning due to its overlap with the effects of temperature and top-$p$ [23].
- **Top-$p$:** A continuous parameter recommended between 0.7 and 0.9. Based on [13], we extend the interval to $[0.6, 1.0]$.
- **Temperature:** A continuous parameter. Based on [4] and [12], values in $[0.1, 2.0]$ are explored. These works found a significant correlation between LLM performance and temperature, with values above 2.0 showing no significant variation.
- **Repetition penalty:** A continuous parameter that controls how much token repetition is penalized during inference. Following [9], the search range is set to $[1.0, 1.5]$.
- **Maximum new tokens:** Defines the upper bound on generated sequence length. The search range is set to $[64, 512]$ tokens, covering outputs from short summaries to extended simplifications [3].

| Parameter | Lower bound | Upper bound |
|---|---|---|
| Temperature | 0.1 | 2.0 |
| Top-$p$ | 0.6 | 1.0 |
| Top-$k$ | 20 | 640 |
| Repetition penalty | 1.0 | 1.5 |
| Maximum tokens | 64 | 512 |

**Table 1: Inference parameter search space for CMA-ES optimization.**

### 4.1 Evaluation Metrics

Text simplification quality is inherently multi-dimensional, requiring the assessment of both structural transformation and semantic preservation. In this work, we employ two complementary metrics: SARI and SBERT.

*SARI.* SARI is a reference-based metric specifically designed for text simplification [29]. Unlike overlap-based metrics such as BLEU [18], SARI explicitly evaluates the quality of three simplification operations: addition, deletion, and keeping.

Given an input sentence $I$, a system output $O$, and a set of reference simplifications $R$, SARI computes the F1-score for each operation and averages them:

$$\text{SARI}(I, O, R) = \frac{1}{3}\left(F1_{\text{add}} + F1_{\text{keep}} + F1_{\text{delete}}\right) \quad (1)$$

The addition component rewards appropriate insertion of simpler lexical items appearing in the references but not in the input; the deletion component evaluates the removal of unnecessary or complex elements; and the keep component measures correct preservation of essential content. SARI ranges from 0 to 100, where higher values indicate better alignment with human simplifications. SARI has been shown to correlate more strongly with human judgments than BLEU in simplification settings [29], and serves as the primary optimization objective in this work.

*SBERT.* To assess semantic preservation, we employ SBERT [22], based on a siamese network architecture that produces semantically meaningful sentence embeddings. Let $\mathbf{e}_O$ and $\mathbf{e}_R$ denote the embeddings of the system output and the reference simplification, respectively. Semantic similarity is computed as:

$$\text{Sim}_{\text{SBERT}} = \frac{\mathbf{e}_O \cdot \mathbf{e}_R}{\|\mathbf{e}_O\|\|\mathbf{e}_R\|} \tag{2}$$

In this work, SBERT similarity is computed between the model output and the human reference simplifications. Under this formulation, both SARI and SBERT are anchored to the same reference set, which justifies the use of SARI as a single optimization objective. SBERT is therefore employed as a secondary monitoring metric to corroborate that configurations optimized for SARI also maintain semantic fidelity to the reference simplifications.

Both metrics are computed over the FEINA corpus [20], which comprises 619 sentence pairs with up to four human reference simplifications per instance, from which the optimization and test subsets are drawn. Prior analysis of reference-based metrics on Spanish simplification datasets has shown that transformation-oriented and semantic-similarity metrics may exhibit divergent behavior depending on the simplification strategy employed [21], further motivating an empirical characterization of the inference-parameter space.

## 4.2 CMA-ES Configuration

*4.2.1 Problem Formulation.* We formulate inference-time parameter tuning as a single-objective black-box optimization problem. Let

$$\mathbf{x} = [\text{temperature},\ \text{top-}p,\ \text{top-}k,\ \text{repetition penalty},\ \text{max tokens}] \tag{3}$$

denote the vector of decoding parameters controlling the language model generation process. Each configuration $\mathbf{x}$ produces simplified text that is evaluated against reference simplifications using SARI. The optimization objective is:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} \ \overline{\text{SARI}}(\mathbf{x}) \tag{4}$$

where $\overline{\text{SARI}}(\mathbf{x})$ denotes the mean SARI score computed over an optimization subset of 100 instances randomly sampled from the FEINA test set, and $\mathcal{X}$ is the bounded search space defined in Table 1. SBERT similarity is recorded for every evaluated configuration but does not influence the optimization process. The best configuration is subsequently evaluated on a separate held-out subset of 124 instances with no overlap with the optimization subset.

*4.2.2 Algorithm and Implementation.* We employ CMA-ES [7], a derivative-free evolution strategy that adapts its search distribution through covariance matrix updates, making it well suited for continuous black-box optimization with moderate dimensionality. CMA-ES iteratively samples candidate solutions from a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \sigma^2\mathbf{C})$, where $\mathbf{m}$ is the distribution mean, $\sigma$ is the global step size, and $\mathbf{C}$ is the covariance matrix. After evaluating each generation, the distribution parameters are updated based on the ranking of candidate fitness values, enabling the algorithm to learn the local landscape geometry without requiring gradient information.

The implementation uses the `CMAES` class from the `pymoo` optimization framework [2], which provides a standardized interface for single-objective evolutionary optimization. Since `pymoo` operates as a minimization framework, the objective function is reformulated as $f(\mathbf{x}) = -\overline{\text{SARI}}(\mathbf{x})$.

*4.2.3 Search Space Normalization and Bound Handling.* The five inference parameters span heterogeneous scales (e.g., temperature in $[0.1, 2.0]$ vs. top-$k$ in $[20, 640]$). To provide a uniform search landscape for CMA-ES, all parameters are internally normalized to the unit hypercube $[0, 1]^5$ using min–max scaling:

$$\tilde{x}_i = \frac{x_i - l_i}{u_i - l_i} \tag{5}$$

where $l_i$ and $u_i$ are the lower and upper bounds of parameter $i$ as specified in Table 1. Candidate solutions are mapped back to the original parameter space before each LLM evaluation. Bound constraints are enforced through the lower and upper bound vectors defined in the `pymoo` problem specification.

Two of the five parameters—top-$k$ and maximum tokens—are inherently discrete. These are handled by allowing CMA-ES to operate in the continuous normalized space and casting the denormalized values to the nearest integer before model evaluation.

*4.2.4 Algorithm Configuration.* Table 2 summarizes the CMA-ES hyperparameters used in this study.

**Table 2: CMA-ES algorithm configuration.**

| Hyperparameter | Value |
|---|---|
| Population size ($\lambda$) | 100 |
| Number of generations | 25 |
| Initial step size ($\sigma_0$) | 0.3 |
| Initial mean ($\mathbf{m}_0$) | Random within $[0, 1]^5$ |
| Search space | $[0, 1]^5$ (normalized) |
| Total function evaluations | 2,500 |
| Optimization subset | 100 instances (random sample) |
| Random seed | Fixed for reproducibility |

The population size of 100 was selected to ensure sufficient exploration of the five-dimensional parameter space while remaining within a feasible computational budget, given that each function evaluation requires a full inference pass of the LLM over the 100-instance optimization subset. Evaluating on a randomly sampled subset rather than the full test set reduces the per-evaluation cost by approximately 6× while preserving sufficient signal to distinguish high-performing configurations. The initial step size $\sigma_0 = 0.3$ provides broad initial exploration relative to the unit-normalized search space. The initial distribution mean $\mathbf{m}_0$ is drawn at random within the normalized bounds, avoiding bias toward any particular region of the parameter space. A fixed random seed ensures reproducibility of the optimization trajectory. The termination criterion is set to 25 generations, yielding a total budget of 2,500 candidate evaluations. A hash-based caching mechanism stores the evaluation result of each parameter configuration, preventing redundant LLM inference when the optimizer revisits previously evaluated regions of the search space.

# 5 Results

## 5.1 Optimization Convergence

Figure 1 presents the optimization trajectory of CMA-ES over 25 generations. The best SARI score improves rapidly during the first nine generations, rising from 40.49 at initialization to 42.20 at generation 9, representing a 4.2% improvement. No further improvement is observed in the remaining 16 generations, indicating that the algorithm effectively converges within the first third of the allocated budget.

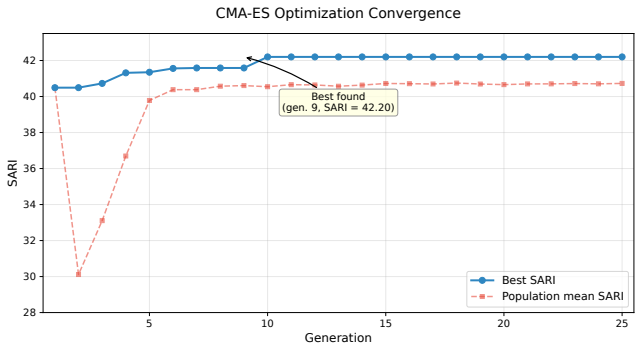



**Figure 1: CMA-ES optimization convergence. The best SARI (solid blue) converges at generation 9. The population means (dashed red) stabilizes around** 40.7**.**

The population means SARI exhibit a characteristic initial drop at generation 2 as CMA-ES samples broadly from the search space, then recover and stabilize around 40.7 by generation 10. The gap between the best and mean values narrows over the course of the optimization, reflecting the concentration of the search distribution around high-performing regions as the covariance matrix adapts.

The rapid convergence suggests that the fitness landscape, while five-dimensional, has a relatively smooth structure with respect to SARI, and that 25 generations with a population of 100 provides a sufficient evaluation budget for this problem. The algorithm identifies near-optimal configurations after only 900 evaluations (9 generations), well within the total budget of 2,500.

Figure 2 shows the evolution of the global step size $\sigma$ over the optimization trajectory. Rather than collapsing toward zero, $\sigma$ remains between 0.19 and 0.30 throughout the 25 generations, indicating that CMA-ES maintains exploratory capacity even after the best solution stabilizes at generation 9. This behavior is consistent with a fitness landscape where the optimum lies in a broad, relatively flat basin rather than a narrow peak.

## 5.2 Best Configuration Analysis

Table 3 presents the best inference configuration identified by CMA-ES, along with the default parameters of the Llama-3.2-1B-Instruct model.

The optimized configuration reveals several noteworthy patterns. The temperature converges to 0.58, remarkably close to the model's default of 0.60, suggesting that the default temperature is already near-optimal for this task. In contrast, top-$p$ is reduced substantially from 0.90 to 0.63, indicating that tighter nucleus truncation benefits

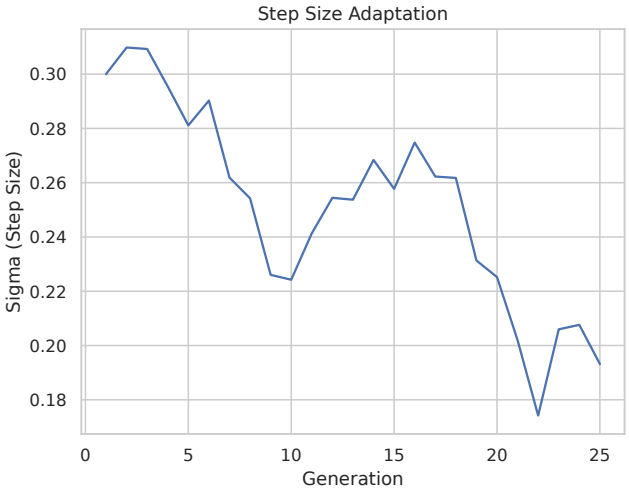



**Figure 2: Evolution of the CMA-ES step size $\sigma$ across generations.**

**Table 3: Default vs. CMA-ES optimized inference configuration. Default values are from the Llama-3.2-1B-Instruct model card and HuggingFace Transformers defaults.**

| Parameter | Default | Optimized |
|---|---|---|
| Temperature | 0.60 | 0.58 |
| Top-$p$ | 0.90 | 0.63 |
| Top-$k$ | 50 | 242 |
| Repetition penalty | 1.00 | 1.00 |
| Maximum tokens | — | 288 |
| **Train SARI** | — | 42.20 |
| Test SARI | $36.57 \pm 0.43^{*}$ | $38.87 \pm 0.45^{*}$ |
| Test SBERT | $0.85 \pm 0.002^{*}$ | $0.88 \pm 0.003^{*}$ |

Note: $p < 0.01$, non parametric hypothesis test

simplification quality by restricting the sampling distribution to higher-probability tokens. Top-$k$ increases from 50 to 242, which, in combination with the reduced top-$p$, provides a wider initial candidate set before nucleus truncation takes effect.

Repetition penalty converges to 1.00 (effectively disabled), consistent with the sensitivity analysis in Section 5.4, which identified this parameter as having the strongest negative impact on SARI. This result suggests that for text simplification, where lexical repetition from the source sentence is expected and desirable (captured by the *keep* component of SARI), penalizing repetition is counterproductive.

## 5.3 Qualitative Analysis

To illustrate the practical impact of the optimized configuration, Table 4 presents two examples from the held-out test subset in which the optimized configuration produces qualitatively better simplifications than the default. These examples were selected from

cases where per-sentence SARI improvements exceeded the overall mean gain, to highlight the types of linguistic patterns that contribute to the observed improvement and are therefore illustrative rather than representative of overall model behavior.

In Example 1, the default reduces the sentence to a verbless fragment by replacing finite verbs ("pertenecen," "son") with participial and adjectival forms ("pertenecientes," "independientes"), which eliminates the main clause structure without reducing linguistic complexity. In contrast, the optimized configuration produces a grammatically simpler construction with finite verbs, thereby eliminating redundancy.

In Example 2, the default replaces the domain-specific term *representación social* with the vague paraphrase "forma de ver el mundo," while the optimized output retains the original terminology and applies targeted lexical simplifications ("cada ser humano" → "cada persona"; "implicaría" → "implica"), as well as a conceptual reduction ("es un arquetipo diferente" → "es diferente").

These patterns are consistent with the optimized parameter configuration: a reduced top-$p$ of 0.63 concentrates sampling on high-probability tokens, and a repetition penalty of 1.00 (effectively disabled) allows the model to reuse source vocabulary without penalization, together favoring the retention of source $n$-grams that also appear in the human references, which is what the *keep* component of SARI rewards [29]. However, the causal link between specific parameter values and the observed linguistic behaviors cannot be established from individual examples alone; the patterns shown here are consistent with, but do not constitute evidence for, the proposed mechanism.

## 5.4 Parameter Sensitivity

To characterize the individual influence of each inference parameter on simplification quality, we conduct a grid search in which each parameter is varied independently while keeping the remaining parameters at their default values. Figures 3–7 present the marginal effect of each parameter on mean SARI.

**Repetition penalty** exhibits the strongest influence on SARI (Figure 3). Mean SARI decreases monotonically from approximately 31 at the lower bound (1.0, i.e., no penalty) to approximately 18 at 1.4. This pronounced negative effect indicates that penalizing token repetition disrupts the model's ability to perform the lexical and syntactic operations that SARI rewards, particularly the *keep* component, which requires preserving content words from the input.

**Temperature** is the second most influential parameter (Figure 4). SARI declines from approximately 26 at low temperatures (0.1) to approximately 19 at the upper bound (2.0). Lower temperatures concentrate the sampling distribution on high-probability tokens, producing outputs that more closely resemble the reference simplifications. As temperature increases, the distribution flattens, introducing lexical diversity that diverges from reference-aligned simplification operations.

**Top-$p$** shows a moderate effect (Figure 5). Mean SARI decreases from approximately 24.2 at $p = 0.6$ to approximately 22.2 at $p = 1.0$, suggesting that tighter nucleus truncation slightly favors simplification quality by restricting sampling to the most probable tokens.

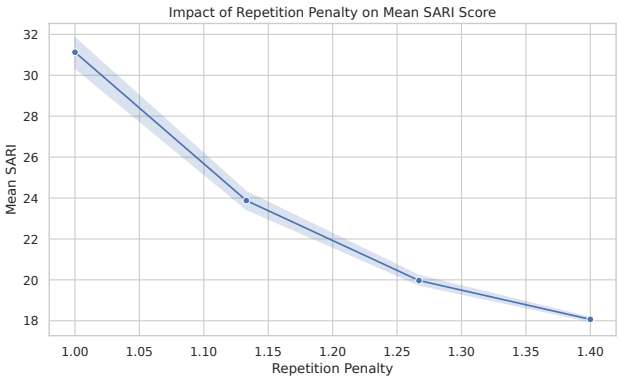



**Figure 3: Impact of repetition penalty on mean SARI score.**

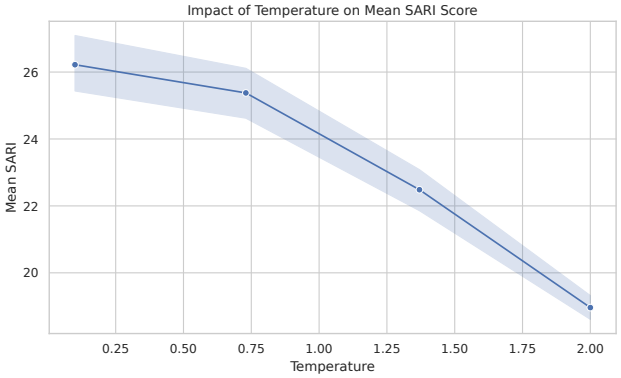



**Figure 4: Impact of temperature on mean SARI score.**

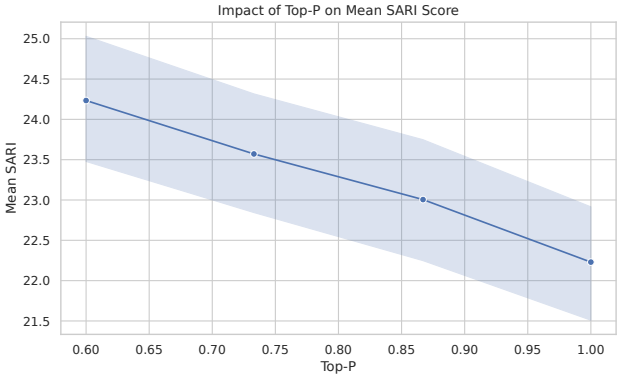



**Figure 5: Impact of top-$p$ on mean SARI score.**

**Top-$k$** and **maximum tokens** exhibit minimal marginal influence (Figures 6 and 7). Top-$k$ shows a slight decrease in SARI from approximately 24 to 23 across its range, while maximum tokens remain nearly flat between 23.0 and 23.4. The limited effect of maximum tokens is expected, as most simplifications in the FEINA

**Table 4: Qualitative comparison of simplified outputs under default and optimized configurations. Bold text highlights key content words from the source that are preserved or lost in each output.**

| | |
|---|---|
| **Example 1:** Grammatical simplification | |
| Source | Están relacionadas entre sí y pertenecen a la misma empresa, pero en definitiva, son independientes. |
| Default | Relacionadas entre sí y pertenecientes a la misma empresa, pero independientes. |
| Optimized | Están en la misma empresa, pero son independientes. |
| **Example 2:** Lexical preservation with targeted simplification | |
| Source | Creo firmemente que la definición de "ser rico" es un arquetipo diferente para cada ser humano, según su **representación social** de lo que ello implicaría. |
| Default | Creo firmemente que la forma de ser rico es diferente para cada persona, según su forma de ver el mundo. |
| Optimized | Creo firmemente que la definición de ser rico es diferente para cada persona según su **representación social** de lo que ello implica. |

corpus are shorter than the lower bound of the search range (64 tokens).

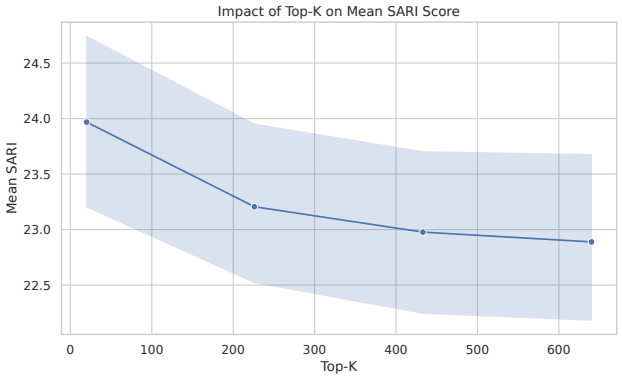



**Figure 6: Impact of top-$k$ on mean SARI score.**

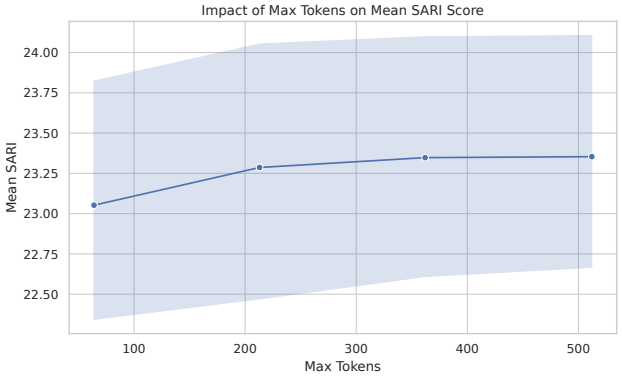



**Figure 7: Impact of maximum tokens on mean SARI score.**

These results confirm that repetition penalty and temperature are the dominant factors in determining simplification quality, consistent with prior findings that decoding temperature significantly affects LLM performance [4, 12] and that repetition penalties influence generation fidelity [9]. The weak influence of top-$k$ corroborates observations that this parameter overlaps functionally with temperature and top-$p$ [23].

## 5.5 Semantic Preservation Validation

A central concern when optimizing a single metric is whether improvements in the target metric come at the expense of other desirable properties. To assess this, we examine the relationship between SARI and SBERT across all configurations evaluated during the CMA-ES optimization.

Figure 8 presents the scatter plot of mean SARI versus mean SBERT for all 2,500 evaluated configurations. The two metrics exhibit a strong positive correlation: configurations that achieve higher SARI scores also tend to preserve semantic similarity, with the highest-performing configurations reaching SARI values of approximately 42 and SBERT scores of approximately 0.89. Conversely, the lowest-performing configurations (SARI $\approx$ 16) also exhibit the lowest semantic similarity (SBERT $\approx$ 0.62).

This positive monotonic relationship suggests that, within the evaluated parameter space and for this model-dataset combination, no clear trade-off between structural transformation quality and semantic preservation is observed.

This finding provides empirical support for the single-objective optimization approach adopted in this work: within the evaluated configurations, maximizing SARI does not compromise SBERT, suggesting that multi-objective optimization may not be necessary for this specific problem setting. The absence of a trade-off between these two metrics suggests that, when both metrics are anchored to the same human reference set, they capture complementary aspects of the same underlying quality dimension rather than competing objectives.

# 6 Discussion

The results demonstrate that CMA-ES effectively identifies inference configurations that improve SARI over the model defaults, converging in only 9 of 25 generations. The optimized configuration confirms that not all inference parameters contribute equally to simplification quality: repetition penalty and temperature dominate, while top-$k$ and maximum tokens have negligible marginal

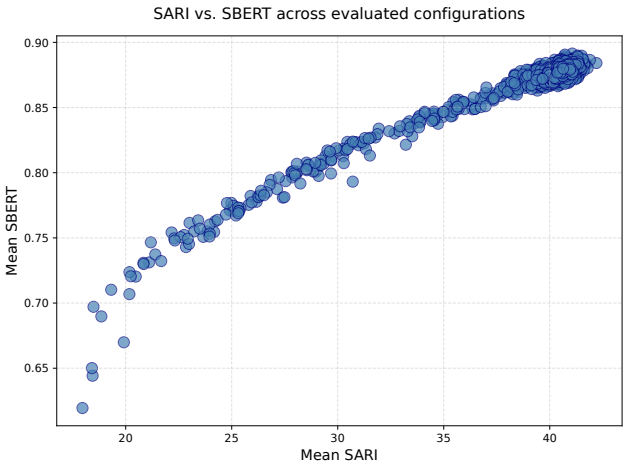



**Figure 8: SARI vs. SBERT across all evaluated configurations. The strong positive correlation suggests that no trade-off between the two metrics is observed within the evaluated parameter space.**

effects. This hierarchy has practical implications for practitioners, as it suggests that tuning efforts should prioritize repetition penalty and top-$p$ over other parameters.

The near-zero optimized repetition penalty challenges the widely adopted recommendation of values around 1.2 [10], indicating that such guidelines do not transfer to TS tasks where content preservation—captured by the *keep* component of SARI—requires repeating words from the source sentence. This finding aligns with observations by Huang et al. [9] that repetition penalties interact differently with generation quality depending on the task.

The optimized temperature of 0.58 is consistent with the ranges identified by Du et al. [4] and Li et al. [12] for tasks requiring high fidelity, and contrasts with the higher values (0.8) recommended for reasoning tasks [13]. The optimized top-$p$ of 0.63 is notably lower than the 0.85–0.98 range found effective for mathematical reasoning [19], reinforcing that inference parameters are task-dependent and cannot be transferred across generative tasks, as previously argued by Wiher et al. [28].

The strong positive correlation between SARI and SBERT (Section 5.5) contrasts with the trade-offs reported by Verma et al. [27] in clinical text summarization, where semantic fidelity and structural quality competed as objectives. This difference likely arises because both metrics in our setting are anchored to the same human reference set, whereas in summarization tasks the objectives may be defined independently. This observation has methodological implications: when evaluation metrics share a common reference, single-objective optimization may be sufficient even when multiple quality dimensions are of interest.

Regarding optimization efficiency, CMA-ES identified the best configuration after only 900 evaluations, compared to the exhaustive grid search approaches employed by Pawar et al. [19], which scale combinatorially with the number of parameters. This confirms the practical advantage of evolutionary strategies for inference parameter tuning, particularly when each evaluation requires expensive LLM inference.

The optimized configuration improves test-set SARI from 36.57 (default) to 38.87, a gain of 2.30 points (6.3%), while SBERT simultaneously increases from 0.85 to 0.88. This concurrent improvement in both metrics on the held-out test subset empirically confirms the finding from Section 5.5: optimizing SARI does not compromise semantic preservation, and in fact improves it.

Several additional limitations should be noted. First, results are specific to a single model (Llama-3.2-1B-Instruct) and dataset (FEINA); transferability to other LLMs, languages, or domains remains to be validated. Second, evaluation relies exclusively on automatic metrics without human judgment. Third, the strong positive correlation between SARI and SBERT may not hold for other datasets or reference configurations.

## 7 Conclusion

We presented an evolutionary optimization framework for tuning inference-time parameters of LLMs applied to Spanish TS. Using CMA-ES to maximize SARI on the FEINA benchmark, we identified an optimized configuration that improves test-set SARI from 36.57 to 38.87 (+6.3%) and SBERT from 0.85 to 0.88, confirming that both metrics benefit from the optimization. The sensitivity analysis revealed that repetition penalty and temperature are the dominant factors, while the strong positive correlation between SARI and SBERT validates the single-objective approach. This work provides a replicable, principled methodology for inference parameter optimization applicable to Spanish and other under-resourced languages.

Future work includes extending the framework to multiple LLMs and datasets, incorporating human evaluation of optimized configurations, exploring adaptive strategies that adjust inference parameters per input rather than using a single static configuration, and evaluating the statistical significance of optimization gains across multiple independent runs with different random seeds.

## 8 Code Repository

The implementation associated with this work, including scripts for data processing, model configuration, and parameter search, is publicly available at https://github.com/dnnxl/llm-param-search.

## 9 Acknowledgment

The authors express their gratitude to the Escuela de Ciencias del Lenguaje and the Vicerrectoría de Investigación y Extensión at the Instituto Tecnológico de Costa Rica (TEC). This research was conducted under the framework of project ID 1470018, titled "Evaluación de modelos de inteligencia artificial en simplificación automática de textos en español para personas con discapacidades visuales." Additionally, the authors acknowledge the computational resources provided through a machine allocation on the Kabré supercomputer at the Costa Rica National High Technology Center and the Scientific Cluster of the University of Costa Rica, which contributed to the completion of this work.

## References


[1] Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier Automatic Sentence Simplification Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, Sebastian Padó and Ruihong Huang (Eds.). Association for Computational Linguistics, Hong Kong, China, 49–54. doi:10.18653/v1/D19-3009

[2] Julian Blank and Kalyanmoy Deb. 2020. pymoo: Multi-Objective Optimization in Python. *IEEE Access* 8 (2020), 89497–89509. doi:10.1109/ACCESS.2020.2990567

[3] Bradley Butcher, Michael O'Keefe, and James Titchener. 2025. Precise length control for large language models. *Natural Language Processing Journal* 11 (2025), 100143.

[4] Weihua Du, Yiming Yang, and Sean Welleck. 2025. Optimizing Temperature for Language Models with Multi-Sample Inference. In *Proceedings of the 42nd International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 267)*. PMLR, Vancouver, Canada, 1–21. https://openreview.net/forum?id=rmWpE3FrHW ICML 2025 Poster.

[5] Isabel Espinosa-Zaragoza, José Abreu-Salas, Elena Lloret, Paloma Moreda, and Manuel Palomar. 2023. A Review of Research-Based Automatic Text Simplification Tools. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing (RANLP 2023)*. INCOMA Ltd., Varna, Bulgaria, 321–330. doi:10.26615/978-954-452-092-2_035

[6] Daniel Ferrés and Horacio Saggion. 2022. ALEXSIS: A Dataset for Lexical Simplification in Spanish. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), Marseille, France, 3582–3594. https://aclanthology.org/2022.lrec-1.383

[7] Nikolaus Hansen. 2006. The CMA Evolution Strategy: A Comparing Review. In *Towards a New Evolutionary Computation: Advances in Estimation of Distribution Algorithms*, José A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea (Eds.). Studies in Fuzziness and Soft Computing, Vol. 192. Springer, Berlin, Heidelberg, 75–102. doi:10.1007/3-540-32494-1_4

[8] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. OpenReview.net, Addis Ababa, Ethiopia, 1–16. https://openreview.net/forum?id=rygGQyrFvH Published as a conference paper at ICLR 2020.

[9] Donghao Huang, Thanh-Son Nguyen, Fiona Liausvia, and Zhaoxia Wang. 2025. RAP: A Metric for Balancing Repetition and Performance in Open-Source Large Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Vol. 1: Long Papers. Association for Computational Linguistics, Albuquerque, New Mexico, USA, 1479–1496. doi:10.18653/v1/2025.naacl-long.82

[10] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858* (Sep 2019), 1–18. arXiv:1909.05858 [cs.CL] doi:10.48550/arXiv.1909.05858

[11] Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023. BLESS: Benchmarking Large Language Models on Sentence Simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Singapore, 13291–13309. doi:10.18653/v1/2023.emnlp-main.822

[12] Lujun Li, Lama Sleem, Geoffrey Nichil, Radu State, et al. 2025. Exploring the impact of temperature on large language models: Hot or cold? *Procedia Computer Science* 264 (2025), 242–251.

[13] Fan Liu, Wenshuo Chao, Naiqiang Tan, and Hao Liu. 2025. Bag of Tricks for Inference-Time Computation of LLM Reasoning. In *Advances in Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks*. Curran Associates, Inc., Vancouver, Canada, 1–17. https://openreview.net/forum?id=c6OvkQTV2D NeurIPS 2025 Dataset and Benchmark Track.

[14] Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares-López, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. 2024. Decoding-time Realignment of Language Models. In *Proceedings of the 41st International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 235)*. PMLR, Vienna, Austria, 31015–31031. https://proceedings.mlr.press/v235/liu24i.html

[15] Mounica Maddela. 2023. *Building and Evaluating Controllable Models for Text Simplification.* Ph. D. Dissertation. Georgia Institute of Technology.

[16] Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A Learnable Evaluation Metric for Text Simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Toronto, Canada, 16383–16408. doi:10.18653/v1/2023.acl-long.896

[17] Alejandro Morales-Hernández, Inneke Van Nieuwenhuyse, and Sebastian Rojas Gonzalez. 2023. A survey on multi-objective hyperparameter optimization algorithms for machine learning. *Artificial Intelligence Review* 56, 8 (2023), 8043–8093.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. doi:10.3115/1073083.1073135

[19] Pranav Pawar, Dhwaj Jain, Varun Gupta, Kaustav Dedhia, Dashrath Kale, and Sudhir Dhekane. 2025. Systematic Optimization of Open Source Large Language Models for Mathematical Reasoning. *arXiv preprint arXiv:2509.07238* (Sep 2025). doi:10.48550/arXiv.2509.07238 Version 1.

[20] Nelson Pérez-Rojas, Saul Calderon-Ramirez, Martín Solís-Salazar, Mario Romero-Sandoval, Monica Arias-Monge, and Horacio Saggion. 2025. A Novel Spanish Dataset for Financial Education Text Simplification Targeting Visually Impaired Individuals. *IEEE Access* 13 (May 2025), 87472–87484. doi:10.1109/ACCESS.2025.3568693 Licensed under CC BY 4.0.

[21] Nelson Pérez-Rojas, Santiago Castrillo-Moncada, Ana Laura Mora-Arias, Saúl Calderón-Ramírez, Martín Solís-Salazar, and Monserrat Ramírez-Castro. 2025. Reference-Based Metric Analysis for Evaluating Spanish Text Simplification. In *Proceedings of the 2025 IEEE 7th International Conference on Bio-Inspired Processing (BIP)*. IEEE, Pérez Zeledón, Costa Rica, 1–6. doi:10.1109/BIP63421.2025.00014 In press. Accepted for publication.

[22] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

[23] Matthew Renze. 2024. The Effect of Sampling Temperature on Problem Solving in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA, 7346–7356. doi:10.18653/v1/2024.findings-emnlp.432

[24] Horacio Saggion, Elena Gómez-Martínez, Esteban Etayo, Alberto Anula, and Lorena Bourg. 2011. Text Simplification in Simplext: Making Texts More Accessible. *Procesamiento del Lenguaje Natural* 47 (Sep 2011), 341–342. https://www.redalyc.org/pdf/5157/515751747047.pdf Recibido 03-05-2011, aceptado 24-05-2011.

[25] Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. Making it Simplext: Implementation and evaluation of a text simplification system for Spanish. *ACM Transactions on Accessible Computing* 6, 4 (2015), 1–36.

[26] Matthew Shardlow et al. 2024. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*. ELRA Language Resource Association, Torino, Italia, 38–46. CC BY-NC 4.0.

[27] Deepika Verma, Daison Darlan, and Rammohan Mallipeddi. 2026. Multi-Objective Optimization of Large Language Model Summaries for Clinical Text Narratives. *IEEE Access* 14 (Jan 2026), 13597–13609. doi:10.1109/ACCESS.2026.3656535

[28] Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On Decoding Strategies for Neural Text Generators. *Transactions of the Association for Computational Linguistics* 10 (2022), 997–1012. doi:10.1162/tacl_a_00502

[29] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics* 4 (2016), 401–415.