Dynamic Profile-Guided Decoding: Inference-Time Personalization for Spanish Lexical Simplification

Author Name(s)

Abstract

Lexical Simplification (LS) typically aims to reduce text complexity for a broad, generic readership, often overlooking the idiosyncratic needs of specific users, such as those with neurocognitive impairments. While Large Language Models (LLMs) have achieved state-of-the-art fluency in simplification tasks, adapting them to individual user profiles traditionally requires computationally expensive fine-tuning. This paper introduces **Dynamic Profile-Guided Decoding (DPGD)**, a framework for Personalized Lexical Simplification (PLS) in Spanish that operates entirely at inference time.

DPGD bypasses model retraining by manipulating the output probability distribution of a frozen LLM during decoding. We propose a *Hybrid Difficulty Profile* (HDP) that combines frequency-based difficulty, morphological complexity, and explicit user knowledge flags into a per-user lexicon of difficult Spanish word types. From this profile, and the LLM's static token-embedding matrix, we derive a *Hybrid Difficulty Embedding Set*: a set of difficulty-weighted semantic prototypes in the model's embedding space. At decoding time, DPGD uses these prototypes to construct a *Semantic Distance Penalty* that down-weights tokens whose embeddings lie close to any difficult prototype, while also applying deterministic penalties based on frequency and morphology.

We evaluate DPGD using the Marco-LLM-ES backbone on the ALEXSIS dataset under simulated user profiles. Our experiments show that DPGD substantially improves adherence to user-specific lexical constraints, as measured by a Profile Hit Rate metric, compared to zero-shot prompting and frequency-only decoding baselines, while maintaining competitive SARI, readability, and BERTScore values. This demonstrates that inference-time personalization via logit manipulation can offer a scalable path toward cognitively accessible text simplification in Spanish without per-user fine-tuning.

1 Introduction

Access to information is a fundamental right, yet the linguistic complexity of essential texts—ranging from legal documents to health instructions—often creates insurmountable barriers for substantial segments of the population. Lexical Simplification (LS) addresses this by replacing complex vocabulary with simpler synonyms to enhance comprehensibility. However, traditional LS systems predominantly operate on a "one-size-fits-all" paradigm, optimizing output for a generic "average" reader. This approach fails to account for the nuance of *cognitive accessibility*: a word considered "simple" by frequency metrics may remain unintelligible to a user with specific neurocognitive conditions, such as Down syndrome or aphasia, while a technical term might be perfectly understood by a specialist with dyslexia.

The emergence of Large Language Models (LLMs) has revolutionized text generation, offering superior fluency compared to rule-based or statistical pipeline approaches. Despite this capability, controlling LLM output to adhere to strict, personalized constraints remains a significant challenge. Standard adaptation methods, such as Supervised Fine-Tuning (SFT) or Parameter-Efficient Fine-Tuning (PEFT), are data-intensive and computationally rigid; they cannot dynamically adjust to a new user profile without retraining.

To bridge the gap between static general simplification and dynamic personalization, we propose **Dynamic Profile-Guided Decoding (DPGD)**. This framework shifts the burden of adaptation from the training phase to the inference phase. By intercepting the decoding loop of a frozen Spanish-centric LLM (Marco-LLM-ES), DPGD manipulates the logit probability distribution in real time. It suppresses the generation of words that are semantically close to a user's *Hybrid Difficulty Profile*—a composite representation that integrates frequency norms, morphological complexity, and explicit knowledge flags—and to nearby concepts in the model's embedding space.

The contributions of this paper are threefold:

1. We formalize the task of Personalized Lexical Simplification (PLS) as a constrained decoding problem rather than a translation task, eliminating the need for massive parallel corpora for every target user profile.
2. We introduce the *Semantic Distance Penalty (SDP)*, a max-pooled embedding-based mechanism that penalizes not just specific forbidden word types, but also tokens whose embeddings are close to any difficulty-weighted prototype derived from the user's profile.
3. We provide an empirical evaluation on the Spanish ALEXSIS dataset under simulated user profiles, demonstrating that DPGD improves personalization alignment (Profile Hit Rate) over zero-shot prompting and frequency-based baselines while preserving strong simplification and semantic adequacy.

The remainder of this paper is organized as follows: Section 2 reviews related work in Spanish LS and constrained decoding. Section 3 formulates the problem. Section 4 details the DPGD architecture, the Hybrid Difficulty Profile, and the associated Hybrid Difficulty Embedding Set. Section 5 describes the experimental setup, and Section 6 presents the results. Section 7 discusses the findings, and Section 8 concludes with directions for future work.

2 Background and Related Work

This section reviews the evolution of Lexical Simplification (LS), specifically within the Spanish language context, and examines the shift from generalized heuristics to personalized, user-centric adaptations. Furthermore, we discuss the technical transition from pipeline-based architectures to Large Language Models (LLMs) and the emerging paradigm of inference-time control via logit manipulation.

2.1 Spanish Lexical Simplification (LS)

Lexical Simplification is the process of identifying complex lexical items in a text and replacing them with simpler semantic equivalents to improve comprehensibility while preserving the original meaning. Historically, Spanish LS has relied on pipeline architectures consisting of three stages: Complex Word Identification (CWI), Substitute Generation (SG), and Substitute Ranking (SR) [1].

Early prominent systems, such as LexSiS [2], utilized static resources including the Spanish Open Thesaurus and frequency counts from the CREA corpus to rank candidates. These systems demonstrated that effective simplification could be achieved without massive parallel corpora by leveraging word length and frequency heuristics. More recently, the field has moved toward neural approaches. The release of datasets like ALEXSIS [3] has standardized benchmarking, providing aligned complex-simple pairs specifically for Spanish. However, these systems predominantly optimize for a "general" simplification standard, often equating "simple" with "high frequency", which fails to address the nuance that a word's difficulty is often idiosyncratic to the user's cognitive profile.

2.2 Personalization and Cognitive Accessibility

While general LS aims to lower the reading level for a broad audience, Personalized Lexical Simplification (PLS) seeks to adapt text to specific user constraints. This is particularly critical for accessible NLP targeting populations with neurocognitive impairments, such as Down syndrome (DS). Research indicates that individuals with DS experience distinct linguistic challenges, including deficits in morphological processing and expressive vocabulary that are not strictly correlated with standard frequency metrics [4, 5].

Current evaluation metrics for simplification, such as SARI [6] or readability formulas like the Szigriszt-Pazos index [7], typically reward generic fluency and structural simplicity. They often fail to capture *Personalized Ease of Comprehension*, where a specific user might require the retention of a low-frequency technical term they know, while needing simplification for high-frequency abstract concepts which they find cognitively dissonant.

2.3 LLMs and Inference-Time Intervention

The advent of Large Language Models (LLMs) has revolutionized text simplification by unifying CWI and substitution into a single generative process. Models adapted for Spanish, such as Marco-LLM-ES [8], demonstrate superior fluency compared to earlier statistical methods. However, controlling the output of an LLM to strictly adhere to a user's difficulty profile usually requires Supervised Fine-Tuning (SFT) or Parameter-Efficient Fine-Tuning (PEFT), both of which are data-intensive and computationally expensive to adapt dynamically for individual users.

To address the rigidity of training-based adaptation, recent research has focused on *constrained decoding* or inference-time intervention. This involves manipulating the model's probability distribution (logits) just prior to token selection. Techniques such as weighted decoding or logit suppression allow for the injection of external constraints—such as avoiding specific words or steering semantics—without altering the model's weights [9].

The methodology proposed in this paper, Dynamic Profile-Guided Decoding (DPGD), builds upon this concept. Unlike standard constrained decoding which often focuses on toxicity reduction or keyword inclusion, DPGD utilizes a *Hybrid Difficulty Profile* and a corresponding *Hybrid Difficulty Embedding Set* to apply negative constraints in the semantic embedding space, directly penalizing tokens that are semantically proximal to concepts the user finds difficult [10].

3 Problem Formulation

We formulate the task of Personalized Lexical Simplification (PLS) not as a standard sequence-to-sequence translation problem trained on large parallel corpora, but as a *constrained decoding* problem for an autoregressive LLM. The objective is to guide a frozen Large Language Model (LLM) to generate text that preserves the semantic content of a source sentence while rigorously adhering to the lexical constraints imposed by a specific user's cognitive profile.

3.1 Task Definition

Let V denote the finite vocabulary of subword tokens used by the LLM tokenizer, and let $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$ denote sequences of tokens in V representing the source and target sentences, respectively. An autoregressive LLM with parameters θ defines a conditional distribution

$$P_\theta(Y \mid X) = \prod_{k=1}^{m} P_\theta\big(y_k \mid X, Y_{<k}\big),$$

where $Y_{<k} = (y_1, \ldots, y_{k-1})$ is the prefix generated up to step $k-1$.

Given a source sentence X in Spanish and a user-specific difficulty profile $\mathcal{P}_u$, the system must generate a target token sequence Y such that, when detokenized, it corresponds to a simplified Spanish sentence. The generation process is governed by the conditional distribution of the LLM, but this distribution is modified at inference time so as to minimize the occurrence of tokens corresponding to word types defined as "difficult" within $\mathcal{P}_u$ or semantically related to such word types.

Let $P_\theta(Y \mid X)$ denote the conditional distribution induced by the frozen LLM. A standard LLM aims to generate a high-likelihood sequence by approximately maximizing $P_\theta(Y \mid X)$. In contrast, the proposed DPGD system seeks to approximate the maximization of a *profile-aware* distribution,

$$P_\theta(Y \mid X, \mathcal{P}_u),$$

implemented by dynamically reshaping the logits at each decoding step in such a way that tokens associated with high user-specific difficulty receive reduced probability mass.

3.2 Inputs and Outputs

The system operates with the following inputs and outputs:

- **Input 1: Source Text (X).** A grammatically complex Spanish sentence containing low-frequency vocabulary, complex morphological structures, or abstract concepts, represented as a token sequence $(x_1, \ldots, x_n)$ in V.

- **Input 2: Hybrid Difficulty Profile ($\mathcal{P}_u$).** A dynamic lexicon representing the user's specific linguistic limitations at the level of surface word types. Let $\mathcal{W}$ denote the set of whitespace-separated word types that can appear in the source or target texts. We represent the profile as a set of tuples mapping difficult Spanish word types w to their associated difficulty attributes:

$$\mathcal{P}_u = \big\{ \big(w, F_{w,\text{norm}}, M_w, \mathcal{K}_w\big) \;\big|\; w \in V_{\text{Difficult}} \big\}, \tag{1}$$

  where $V_{\text{Difficult}} \subseteq \mathcal{W}$ is the subset of word-type vocabulary (surface word forms) that is known to be challenging for the user, $F_{w,\text{norm}}$ is a normalized frequency-based difficulty score, M_w is a morphological complexity score, and $\mathcal{K}_w$ is a known/unknown flag. Embeddings associated with these word types are derived from the LLM as described below, and are not considered intrinsic components of $\mathcal{P}_u$.

- **Output: Target Text (Y).** A simplified Spanish sentence that conveys (as closely as possible) the same meaning as X but utilizes a vocabulary and structure compliant with the constraints encoded in $\mathcal{P}_u$, represented as a token sequence $(y_1, \ldots, y_m)$ in V.

3.3 Notation and Formal Framework

To describe the Dynamic Profile-Guided Decoding (DPGD) methodology, we establish the following notation:

- **Token Vocabulary (V) and Word Types ($\mathcal{W}$).** V is the set of all subword tokens recognized by the LLM tokenizer, with vocabulary size $|V|$. $\mathcal{W}$ is the set of word types (whitespace-separated surface forms) that can appear in the source or target texts and in $\mathcal{P}_u$.

- **Tokenizer Decomposition of Words.** For each $w \in \mathcal{W}$, let

$$\text{Tok}(w) = \big(\tau_1^{(w)}, \ldots, \tau_{L(w)}^{(w)}\big), \qquad \tau_j^{(w)} \in V,$$

  denote the unique sequence of tokenizer tokens that produces the surface form w when detokenized in isolation.

- **Tokens and Indices.** We denote by $t_i \in V$ the i-th token in the vocabulary. At decoding step k, the LLM has produced a prefix $Y_{<k} = (y_1, \ldots, y_{k-1})$ of output tokens and assigns a logit $z_i^{(k)}$ to each possible next token t_i.

- **Context-Dependent Token–Word Alignment.** Because we use a subword tokenizer (e.g., BPE or SentencePiece), there is in general no one-to-one mapping from token types to word types. At decoding step k, we maintain the sequence of tokens that form the current surface word suffix, i.e., the tokens produced since the last whitespace token. Given this suffix and a candidate next token t_i, we define

$$\mathcal{A}^{(k)}(t_i) = \big\{\, w \in V_{\text{Difficult}} \;\big|\; \text{Tok}(w) \text{ has the current suffix followed by } t_i \text{ as a prefix} \,\big\}.$$

  We then define a context-dependent alignment function

$$\phi^{(k)} : V \to \mathcal{W} \cup \{\varnothing\}, \qquad w_i^{(k)} = \phi^{(k)}(t_i) = \begin{cases} \arg\max\limits_{w \in \mathcal{A}^{(k)}(t_i)} S_w, & \text{if } \mathcal{A}^{(k)}(t_i) \neq \varnothing, \\ \varnothing, & \text{otherwise,} \end{cases}$$

  where S_w is the scalar difficulty score defined below. When $w_i^{(k)} = \varnothing$, the token t_i has no active word-level difficulty entry at step k. To keep notation light, we drop the superscript (k) when the decoding step is clear and simply write $w_i = \phi(t_i)$.

- **Logits (z) and Probabilities (p).** At decoding step k, the LLM produces a raw, unnormalized logit vector $\mathbf{z}^{(k)} \in \mathbb{R}^{|V|}$, where $z_i^{(k)}$ is the logit associated with token t_i given the current decoding history. The token probability distribution is obtained by applying a temperature-scaled softmax function to the logits:

$$\mathbf{p}^{(k)} = \text{softmax}\big(\mathbf{z}^{(k)}/\tau\big), \tag{2}$$

  where $\tau > 0$ is a temperature parameter.

- **Embeddings.** Let $e_i \in \mathbb{R}^d$ denote the LLM embedding of token t_i (a row of the model's static token embedding matrix, typically tied with the language modeling head). For each word type $w \in \mathcal{W}$, we define its embedding $e_w \in \mathbb{R}^d$ by averaging the embeddings of its tokenizer decomposition:

$$e_w = \frac{1}{L(w)} \sum_{j=1}^{L(w)} e_{\tau_j^{(w)}}.$$

  This guarantees that both token embeddings e_i and word embeddings e_w live in the same vector space.

- **Penalty Vector (P).** At each decoding step, DPGD computes a penalty vector $\mathbf{P}^{(k)} \in \mathbb{R}^{|V|}$ whose i-th component $P_i^{(k)}$ depends on the user profile $\mathcal{P}_u$ and the embedding-based difficulty representation. The LLM logits are then transformed into modified logits $\mathbf{z}'^{(k)}$ via

$$\mathbf{z}'^{(k)} = \mathbf{z}^{(k)} - \lambda \cdot \mathbf{P}^{(k)}, \tag{3}$$

  where $\lambda \geq 0$ is a global *personalization strength* hyperparameter. Sampling (or beam search) is then performed from the temperature-scaled softmax $\text{softmax}(\mathbf{z}'^{(k)}/\tau)$.

- **Profile Components.** For each $w \in V_{\text{Difficult}}$, the profile stores:
  - $F_{w,\text{norm}} \in [0, 1]$: a normalized lexical frequency–based difficulty score, with larger values corresponding to rarer (more difficult) words.
  - $M_w \in [0, 1]$: a morphological complexity score (binary or scaled).
  - $\mathcal{K}_w \in \{0, 1\}$: a binary "unknown-to-the-user" flag used for hard masking.
  - S_w: a composite scalar difficulty score used in both deterministic penalties and the semantic component:

$$S_w = \beta_{\text{Freq}} F_{w,\text{norm}} + \beta_{\text{Morph}} M_w + \beta_{\text{Mask}} \mathcal{K}_w, \tag{4}$$

    with non-negative weights $\beta_{\text{Freq}}, \beta_{\text{Morph}}, \beta_{\text{Mask}}$.
  - E_{Diff}: the *Hybrid Difficulty Embedding Set*, defined as the set of normalized, weighted difficulty prototypes

$$E_{\text{Diff}} = \left\{ (S_w, \tilde{e}_w) \;\middle|\; w \in V_{\text{Difficult}} \right\}, \qquad \tilde{e}_w = \frac{e_w}{\|e_w\|_2}. \tag{5}$$

    Rather than compressing the profile into a single centroid vector, the Semantic Distance Penalty (SDP) operates on this set by comparing each candidate token to its *closest* difficult prototype (see Section 4.3).

3.4 Assumptions and Constraints

The formulation of the DPGD prototype relies on the following assumptions:

1. **Frozen Backbone.** The weights of the foundation model (e.g., Marco-LLM-ES) remain static. No gradient updates (SFT/PEFT) are performed.

2. **Logit Access.** The inference environment permits interception and modification of the generation pipeline prior to sampling (specifically via a `LogitsProcessor`-style interface that operates before temperature scaling and softmax).

3. **Language Specificity.** The linguistic heuristics (morphology and frequency norms) are specifically calibrated for the Spanish language.

4. **Token–Word Alignment.** We assume that we can compute, at inference time, the context-dependent alignment function $\phi^{(k)}$ described above by tokenizing every word $w \in \mathcal{P}_u$ with the same tokenizer used by the LLM. This allows the system to handle multi-token difficult words under BPE or SentencePiece tokenization without incorrectly penalizing unrelated occurrences of shared subword tokens.

5. **Empty Profile Fallback.** When $V_{\text{Difficult}} = \varnothing$ (i.e., the user profile does not currently mark any word types as difficult), the Semantic Distance Penalty and all profile-based terms are defined to be zero, so that DPGD reduces exactly to the baseline LLM decoding.

4 Methodology

In this section, we present the Dynamic Profile-Guided Decoding (DPGD) framework. DPGD is designed as a modular prototype to achieve Personalized Lexical Simplification (PLS) without the computational overhead of model fine-tuning. The personalization logic is shifted entirely to the inference phase, where a user's specific difficulty profile is used to manipulate the Large Language Model's (LLM) output distribution in real time.

4.1 Overall System Architecture

The DPGD framework operates as an interceptor within the standard LLM generation pipeline. It consists of three interconnected modules that function sequentially during inference:

1. **Hybrid Difficulty Profile (HDP) Manager.** This component acts as the data ingestion layer. It fuses discrete psycholinguistic statistics (such as word frequency and morphological complexity) with explicit knowledge flags to construct the personalized profile $\mathcal{P}_u$.

2. **Semantic Analyzer.** This module operates on the Hybrid Difficulty Profile $\mathcal{P}_u$ and its associated embeddings to abstract the user's difficulties into a high-dimensional vector space. It constructs the *Hybrid Difficulty Embedding Set* E_{Diff} (Equation 5), a set of normalized, difficulty-weighted prototypes that encode which regions of the embedding space should be suppressed for the user.

3. **Profile-Linked LogitsProcessor ($\mathcal{L}_{\mathbf{PLS}}$).** This is the execution core of the system. Implemented as a custom hook in the decoding loop, it calculates a composite penalty vector $\mathbf{P}^{(k)}$ at each step and applies it to the raw logits $\mathbf{z}^{(k)}$ via Equation 3 before temperature scaling and softmax are applied (Equation 2).

4.2 Design of the Hybrid Difficulty Profile ($\mathcal{P}_u$)

The foundation of the personalization mechanism is the Hybrid Difficulty Profile (HDP). Unlike traditional systems that rely solely on frequency counts, the HDP incorporates a multi-dimensional view of difficulty tailored to the Spanish language and specific neurocognitive profiles (e.g., Down syndrome).

As introduced in Section 3, the profile $\mathcal{P}_u$ maps specific Spanish word types to a set of scalar attributes. For clarity, we view $\mathcal{P}_u$ as comprising two conceptual layers.

4.2.1 Layer 1: Deterministic Metrics

These are pre-computable scalar scores derived from linguistic resources:

- **Lexical Frequency Score ($F_{w,\mathbf{norm}}$).** Calculated based on Spanish psycholinguistic norms (e.g., Duchon et al.). Since rarity correlates with difficulty, $F_{w,\text{norm}}$ is normalized to $[0, 1]$ such that a higher score indicates lower frequency (higher difficulty).

- **Morphological Complexity Score (M_w).** A binary or scaled score flagging words with complex structural features (e.g., irregular verbs, high derivational complexity). This is critical for users with specific impairments in grammatical processing.

- **Known/Unknown Status ($\mathcal{K}_w$).** A binary flag derived from explicit user feedback or vocabulary testing, where $\mathcal{K}_w = 1$ indicates a word the user definitely cannot comprehend and should therefore be hard-masked.

Given non-negative weights $\beta_{\text{Freq}}, \beta_{\text{Morph}}, \beta_{\text{Mask}}$, we combine these metrics into a composite scalar difficulty score S_w as defined in Equation 4:

$$S_w = \beta_{\text{Freq}} F_{w,\text{norm}} + \beta_{\text{Morph}} M_w + \beta_{\text{Mask}} \mathcal{K}_w.$$

Although $F_{w,\text{norm}}$ and M_w will also appear directly in the final penalty (Section 4.4), maintaining S_w as a separate aggregate allows us to differentially scale their impact in semantic versus deterministic components.

4.2.2 Layer 2: Semantic Representation (E_{Diff})

To address semantically related difficulties, we represent the user's "difficulty region" in the LLM embedding space as a finite set of normalized, weighted prototypes. For each $w \in V_{\text{Difficult}}$ we compute the normalized word embedding

$$\tilde{e}_w = \frac{e_w}{\|e_w\|_2},$$

where e_w is defined from token embeddings in Section 3.3, and store the pair $(S_w, \tilde{e}_w)$. The resulting *Hybrid Difficulty Embedding Set* is precisely the set E_{Diff} defined in Equation 5:

$$E_{\text{Diff}} = \big\{ (S_w, \tilde{e}_w) \;\big|\; w \in V_{\text{Difficult}} \big\}.$$

Crucially, DPGD does *not* collapse these prototypes into a single centroid vector. Instead, the Semantic Distance Penalty (SDP) introduced in Section 4.3 compares each candidate token to its *closest* difficult prototype. This implements a set-based, max-pooled notion of difficulty: a token is penalized if it is close to *any* difficult concept, rather than to the average of all difficult concepts.

4.3 Semantic Distance Penalty (SDP)

A core contribution of this work is the Semantic Distance Penalty (SDP). This mechanism ensures that the model does not simply substitute a low-frequency difficult word with a higher-frequency word that conveys the same complex concept, and it avoids the "centroid fallacy" inherent in averaging semantically disjoint difficult words.

For every candidate token t_i in the LLM's vocabulary during decoding, we associate an embedding e_i (the corresponding token embedding) and its ℓ_2-normalized version

$$\tilde{e}_i = \frac{e_i}{\|e_i\|_2}.$$

We then compute its similarity to each profile prototype $\tilde{e}_w$ and define the maximum similarity. Let

$$\text{CosineSim}(a, b) = \frac{a^\top b}{\|a\|_2 \, \|b\|_2}$$

denote cosine similarity. Since both $\tilde{e}_i$ and $\tilde{e}_w$ are normalized, this reduces to the dot product $\tilde{e}_i^\top \tilde{e}_w$. We define

$$s_i(w) = \text{CosineSim}(\tilde{e}_i, \tilde{e}_w), \qquad s_i^{\max} = \max_{w \in V_{\text{Difficult}}} s_i(w), \tag{6}$$

with corresponding nearest difficult word

$$w^\star(e_i) = \arg \max_{w \in V_{\text{Difficult}}} s_i(w).$$

We convert this into a cosine-based distance

$$D_{\min}(e_i) = 1 - s_i^{\max}. \tag{7}$$

Since cosine similarity lies in $[-1, 1]$, the distance $D_{\min}(e_i)$ is in $[0, 2]$, with smaller values corresponding to tokens that are semantically closer to *some* user-specific difficult concept.

We transform this distance into a penalty score $P_{\text{SDP}}(t_i)$ using a distance margin $\delta \in (0, 2]$, a scaling factor $\gamma \geq 0$, and the scalar difficulty of the nearest prototype:

$$P_{\text{SDP}}(t_i) = \gamma \, S_{w^\star(e_i)} \, \max\big(0, \delta - D_{\min}(e_i)\big). \tag{8}$$

Tokens whose embeddings lie within the margin ($D_{\min}(e_i) < \delta$) receive a positive semantic penalty that grows as they become closer to at least one difficult prototype and as the corresponding $S_{w^\star(e_i)}$ increases. Tokens with $D_{\min}(e_i) \geq \delta$ are unaffected by the SDP term.

By convention, when $V_{\text{Difficult}} = \varnothing$ we define $P_{\text{SDP}}(t_i) = 0$ for all tokens t_i. This ensures that in the absence of any profile information the SDP contribution vanishes and DPGD reduces to the baseline model.

This construction is equivalent to a max-pooling operation over the set of similarity scores $\{s_i(w)\}_{w \in V_{\text{Difficult}}}$: the penalty depends only on the *most* similar difficult concept. As a result, tokens that lie near the geometric average of several mutually unrelated difficult concepts but are not particularly close to any one of them receive a much smaller penalty than the actual difficult words themselves.

4.4 Profile-Linked Logit Adjustment ($\mathcal{L}_{\text{PLS}}$)

The final stage is the integration of the semantic penalty with the deterministic scalar penalties to form the total penalty vector $\mathbf{P}^{(k)}$ used to adjust the logits.

For each candidate token t_i at decoding step k, we consider the context-dependent alignment $w_i^{(k)} = \phi^{(k)}(t_i)$ defined in Section 3.3. When $w_i^{(k)} \in V_{\text{Difficult}}$, the composite penalty $P(t_i)$ is given by:

$$P(t_i) = \alpha_{\text{SDP}} \cdot P_{\text{SDP}}(t_i) + \alpha_{\text{Freq}} \cdot F_{w_i,\text{norm}} + \alpha_{\text{Morph}} \cdot M_{w_i} + C_{\text{mask}} \cdot \mathbf{1}[\mathcal{K}_{w_i} = 1]\,, \tag{9}$$

where $\alpha_{\text{SDP}}, \alpha_{\text{Freq}}, \alpha_{\text{Morph}} \geq 0$ are weighting hyperparameters, $C_{\text{mask}} \gg 0$ is a large constant, and $\mathbf{1}[\cdot]$ is the indicator function. When $w_i = \varnothing$, the scalar attributes are treated as zero:

$$F_{w_i,\text{norm}} = 0, \quad M_{w_i} = 0, \quad \mathcal{K}_{w_i} = 0,$$

so that only the semantic distance penalty term $P_{\text{SDP}}(t_i)$ contributes. In practice, setting $\alpha_{\text{Freq}} = \alpha_{\text{Morph}} = 0$ recovers a purely semantic constraint, while setting $\alpha_{\text{SDP}} = 0$ yields a frequency-only baseline.

The hard-masking term is implemented by choosing C_{mask} large enough that, after scaling by λ, the corresponding logit becomes effectively $-\infty$, ensuring that tokens marked with $\mathcal{K}_{w_i} = 1$ are never sampled. In practice, this is realized by explicitly setting those logits to a very negative value.

Given the per-token penalties $P(t_i)$, the modified logits at decoding step k are obtained by applying the vector update in Equation 3 component-wise:

$$z_i^{(k)\prime} = z_i^{(k)} - \lambda \cdot P(t_i).$$

The final token probabilities are then obtained by dividing by the temperature τ and applying softmax as in Equation 2:

$$p_i^{(k)} = \frac{\exp\left(z_i^{(k)\prime}/\tau\right)}{\sum_{j=1}^{|V|} \exp\left(z_j^{(k)\prime}/\tau\right)}.$$

4.5 Implementation Details

The DPGD system is implemented using Python and the HuggingFace Transformers library. Key implementation aspects are:

- **LLM Backbone.** We utilize **Marco-LLM-ES-7B**, a foundation model specifically enhanced for the Spanish language through extensive pretraining on Spanish-specific corpora. This ensures a higher baseline of fluency and morphological correctness compared to general multilingual models.
- **Logit Integration.** The $\mathcal{L}_{\text{PLS}}$ mechanism is implemented via the `LogitsProcessor` API. At each decoding step, the model produces logits $\mathbf{z}^{(k)}$; DPGD computes a penalty vector $\mathbf{P}^{(k)}$ according to Equation 9, forms adjusted logits $\mathbf{z}'^{(k)}$ via Equation 3, and then the standard generation pipeline applies temperature scaling and softmax as in Equation 2.
- **Engineering Optimizations.** To minimize inference latency—a critical requirement for real-time applications—the normalized difficulty prototypes $\{(S_w, \tilde{e}_w)\}_{w \in V_{\text{Difficult}}}$ and the L2-normalized token embedding matrix are pre-computed once at the start of the session. During generation, the semantic distance calculation utilizes vectorized tensor operations in PyTorch: the matrix of normalized token embeddings for the full vocabulary is multiplied by the matrix of normalized difficult prototypes, and the max over prototypes is taken for each token. This avoids slow iterative loops over the vocabulary and yields $O(|V| \cdot |V_{\text{Difficult}}|)$ complexity per step with efficient GPU utilization.
- **Tokenization Alignment.** We do not assume a one-to-one mapping from token types to difficult word types. Instead, we obtain $\text{Tok}(w)$ for every $w \in \mathcal{P}_u$ by running the same tokenizer used by the LLM. At inference time, we maintain the sequence of tokens since the last whitespace and use it to compute the context-dependent alignment set $\mathcal{A}^{(k)}(t_i)$ and the mapping $\phi^{(k)}(t_i)$ defined in Section 3.3. As a result, subword tokens such as "constit" are only treated as difficult when they actually occur as part of a difficult word like "inconstitucionalmente", and not when they appear in unrelated lexical contexts.

- **Hyperparameters.** Key parameters for the prototype include the global strength λ (calibrated relative to the generation temperature τ), the similarity threshold encoded by δ, the scaling factor γ, and the weights $\alpha_{\text{SDP}}, \alpha_{\text{Freq}}, \alpha_{\text{Morph}}$ and $\beta_{\text{Freq}}, \beta_{\text{Morph}}, \beta_{\text{Mask}}$. Unless otherwise stated, we treat the β-weights as part of the definition of the profile and keep them fixed across experiments, while λ, δ, γ, and the α-weights are tuned on a held-out validation split (see Section 5).

5 Experimental Setup

To validate the efficacy of the Dynamic Profile-Guided Decoding (DPGD) framework, we designed an experimental protocol focused on two primary objectives: (i) demonstrating effective lexical simplification in Spanish, and (ii) showing measurably superior personalization compared to non-semantic baselines. This section details the dataset, user profile simulations, baseline configurations, and the evaluation metrics employed.

5.1 Dataset Selection: ALEXSIS

We utilize the **ALEXSIS** dataset (A Dataset for Lexical Simplification in Spanish) [3] as the primary testbed. This dataset was selected because it provides:

- A robust collection of complex Spanish sentences aligned with human-annotated simplifications.
- A "gold standard" reference that allows for the calculation of standard benchmarks like SARI.
- Sufficient linguistic complexity to activate the mechanisms of our difficulty profiles.

We follow the official train/dev/test split provided with ALEXSIS. The LLM backbone is *never* fine-tuned on any split; all experiments use the frozen Marco-LLM-ES-7B model. The development split is used exclusively for hyperparameter selection (e.g., λ, α_{SDP}, α_{Freq}, α_{Morph}, δ, γ), while the test split is reserved for final evaluation.

Preprocessing involves standard tokenization using the Marco-LLM-ES tokenizer. No gradient-based training or parameter updates are performed; all adaptations occur at inference time through the DPGD logit processor.

5.2 Profile Simulation: Synthetic User Groups

A core challenge in evaluating Personalized Lexical Simplification (PLS) is the variability of individual needs. To test the system's adaptability, we simulate three distinct user groups, each represented by a specific configuration of the Hybrid Difficulty Profile ($\mathcal{P}_u$) and weighting hyperparameters:

1. **Group A (Older Adults).**
   - *Profile Characteristics:* This profile is heavily biased toward difficulties with low-frequency technical and legal terminology.
   - *Configuration:* High weight on frequency penalization (α_{Freq}). The difficulty embedding set E_{Diff} is dominated by abstract and specialized vocabulary, so the SDP term actively suppresses their near-synonyms.
2. **Group B (Neurocognitive/Down Syndrome).**
   - *Profile Characteristics:* This profile models users with specific cognitive impairments who struggle with morphological complexity and semantic ambiguity.
   - *Configuration:* High weights on morphological complexity (α_{Morph}) and semantic distance (α_{SDP}). The set of difficult embeddings in E_{Diff} is critical here to suppress conceptually ambiguous terms even if they are high-frequency.
3. **Group C (Control/General).**
   - *Profile Characteristics:* A non-personalized profile based solely on general population statistics.

- *Configuration:* The profile targets only statistically rare words without semantic or morphological bias, mimicking a traditional, "one-size-fits-all" simplification system.

For each synthetic group, we instantiate the difficulty profile $\mathcal{P}_u$ by combining corpus-based frequency norms, morphology-based heuristics, and manually curated seed lists. Concretely, we derive $F_{w,\text{norm}}$ from Spanish psycholinguistic frequency norms, compute M_w using a rule-based morphological analyzer (flagging, e.g., irregular verbs and heavily derived forms), and set $\mathcal{K}_w$ based on group-specific seed vocabularies (e.g., legal terminology for Group A, abstract and morphologically complex items for Group B). The difficult vocabulary subset $V_{\text{Difficult}}$ for each group is then obtained by thresholding on $F_{w,\text{norm}}$ and M_w and augmenting with the corresponding seed lists. The resulting normalized difficulty embeddings $\{(S_w, \tilde{e}_w)\}$ constitute the Hybrid Difficulty Embedding Set E_{Diff} defined in Section 4.2.2.

5.3 Baselines

We compare the performance of the full DPGD system against three distinct baselines to isolate the contributions of the proposed methodology:

1. **Zero-Shot LLM (General Simplification).** The backbone model, *Marco-LLM-ES-7B*, is queried with a standard prompt: *"Simplify this Spanish text for a user with limited reading experience."* This establishes the model's inherent, generalized simplification capability without logit intervention. All decoding parameters (temperature, top-k, top-p, maximum length, etc.) are identical to those used in the DPGD runs.

2. **Frequency-Only Penalization (Ablation Baseline).** The DPGD system is configured with $\alpha_{\text{SDP}} = 0$ and $\alpha_{\text{Morph}} = 0$. This configuration tests whether simple scalar frequency scores alone are sufficient, effectively replicating the logic of traditional pipeline systems (like LexSiS) but integrated into the logit space. The global strength λ and other decoding parameters are tuned on the dev split and then held fixed on the test split.

3. **ALEXSIS Reference.** The human-annotated simplifications provided in the dataset serve as the topline "gold standard" for content preservation and quality.

5.4 Ablation Studies and Hyperparameter Sweep

To rigorously quantify the contribution of the Semantic Distance Penalty (SDP), we conduct ablation runs where we vary which components of the penalty are active:

- **Run A (Frequency-Only).** As described in the baselines, $\alpha_{\text{SDP}} = 0$ and $\alpha_{\text{Morph}} = 0$.
- **Run B (Semantic-Only).** We set $\alpha_{\text{Freq}} = 0$ and $\alpha_{\text{Morph}} = 0$ to test the hypothesis that semantic proximity alone drives difficulty for specific cognitive profiles.
- **Run C (Full DPGD).** The optimal weighted combination of all α terms is used ($\alpha_{\text{SDP}}, \alpha_{\text{Freq}}, \alpha_{\text{Morph}} > 0$).

For each configuration, we perform a sensitivity sweep over the global personalization strength hyperparameter $\lambda \in \{0.5, 1.0, 2.0, 5.0\}$ and, where relevant, over the distance margin δ and the scaling factor γ. Unless otherwise noted, the β-weights are kept fixed across all groups, while $(\lambda, \delta, \gamma, \alpha_{\text{SDP}}, \alpha_{\text{Freq}}, \alpha_{\text{Morph}})$ are tuned separately for each synthetic user group on that group's portion of the ALEXSIS development split. The configuration that maximizes SARI subject to a minimum threshold on Profile Hit Rate (see below) is selected for evaluation on the test set.

5.5 Evaluation Metrics

Evaluation uses a multi-faceted approach that assesses content quality, structural readability, semantic adequacy, and the system's mechanical compliance with user constraints.

We refine the **Profile Hit Rate (PHR)** to be sensitive both to difficult vocabulary present in the source and to difficult vocabulary newly introduced by the system, while ignoring examples where no difficult words are relevant. Let $\text{WordSet}(X)$ denote the set of word types (in $\mathcal{W}$) that appear

Table 1: Quantitative Evaluation Metrics

Metric Category	Specific Metric	Significance
Simplification Quality	**SARI Score**	Measures the quality of lexical substitution, additions, and deletions relative to the reference, balancing simplification and meaning preservation.
Readability (Structure)	**Szigriszt-Pazos Index (ILSP)**	A Spanish-adapted Flesch formula that quantifies structural complexity based on syllables per word and sentence length.
Semantic Adequacy / Fluency Proxy	**BERTScore (F1)**	Measures semantic similarity between system output and reference; used as a proxy to ensure the output remains semantically faithful and generally fluent.
Personalization	**Profile Hit Rate (PHR)**	A custom metric that quantifies how well the model avoids user-marked difficult vocabulary in the output.

in a sentence X after whitespace tokenization. Given a difficulty profile $\mathcal{P}_u$ with difficult word types $V_{\text{Difficult}} \subseteq \mathcal{W}$, a source sentence X, and a generated output sequence Y, we define

$$V_{\text{active}}(X, Y, \mathcal{P}_u) = V_{\text{Difficult}} \cap \big(\text{WordSet}(X) \cup \text{WordSet}(Y)\big), \tag{10}$$

$$V_{\text{violated}}(Y, \mathcal{P}_u) = V_{\text{Difficult}} \cap \text{WordSet}(Y). \tag{11}$$

That is, V_{active} contains the difficult word types that are relevant to the example because they appear either in the source or in the system output, and V_{violated} contains the subset that actually appears in the output.

For any example with $|V_{\text{active}}(X, Y, \mathcal{P}_u)| > 0$, we define the sentence-level PHR as

$$\text{PHR}(X, Y, \mathcal{P}_u) = 1 - \frac{\big|V_{\text{violated}}(Y, \mathcal{P}_u)\big|}{\big|V_{\text{active}}(X, Y, \mathcal{P}_u)\big|}. \tag{12}$$

By construction, $\text{PHR}(X, Y, \mathcal{P}_u) \in [0, 1]$ and equals 1 when the system completely avoids all relevant difficult word types (including those in the source), decreasing linearly as more of those types appear in the simplified output. Note that this metric operates at the *type* level: multiple occurrences of the same difficult word type in Y count as a single violation.

Corpus-level PHR for a given profile is obtained by averaging over examples where difficult words are active:

$$\text{PHR}_{\text{corpus}}(\mathcal{P}_u) = \frac{1}{|\mathcal{I}|} \sum_{j \in \mathcal{I}} \text{PHR}(X_j, Y_j, \mathcal{P}_u), \tag{13}$$

where $\mathcal{I}$ is the set of indices j such that $|V_{\text{active}}(X_j, Y_j, \mathcal{P}_u)| > 0$. Examples for which the profile is irrelevant (no difficult words in source or output) are excluded from the average, so that corpus-level PHR reflects personalization behaviour only where it is meaningful.

6 Results

Present the results clearly using tables and figures. Highlight:

- Quantitative performance comparisons
- Ablation studies
- Sensitivity analyses

7 Discussion

Interpret the results from a critical perspective. Discuss:

- Strengths and limitations of the proposed approach
- Observed trade-offs
- Error analysis, if relevant

8 Conclusion and Future Work

Summarize the key insights, contributions, and results. Discuss potential extensions or next steps for the research.

References

[1] Matthew Shardlow. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70, 2014.

[2] Stefan Bott, Horacio Saggion, and Simon Mille. Can spanish be simpler? lexsis: Lexical simplification for spanish. In *Proceedings of COLING 2012*, pages 357–374. The COLING 2012 Organizing Committee, 2012.

[3] Daniel Ferrés, Horacio Saggion, and Xavier Gómez Guinovart. Alexsis: A dataset for lexical simplification in spanish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2625–2633. European Language Resources Association, 2022.

[4] Anne E Fowler. *Early grammatical development in Spanish children with Down syndrome*. Cambridge University Press, 1988.

[5] Miguel Galeote, Pilar Soto, Eugenia Sebastián, Rocío Rey, and E Checa. The acquisition of productive vocabulary in spanish children with down syndrome. *Journal of Intellectual Disability Research*, 52(5):424–434, 2008.

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

[7] Francisco Szigriszt Pazos. *Sistemas predictivos de legibilidad del mensaje escrito: fórmula de perspicuidad*. Universidad Complutense de Madrid, 1993.

[8] AIDC-AI. Marco-llm-es: A foundation model for spanish, 2023.

[9] Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, 2021.

[10] Thinking Sand. Embedding similarity explained: How to measure text semantics, 2023.