Uniandes at TSAR 2025 Shared Task: Multi-Agent CEFR Text Simplification with Automated Quality Assessment and Iterative Refinement

Felipe Arias-Russi[1,2], **Kevin Cohen-Solano**[1], **Rubén Manrique**[1]
[1]Systems and Computing Engineering Department, Universidad de los Andes, Colombia
[2]Department of Mathematics, Universidad de los Andes, Colombia

{af.ariasr, k.cohen, rf.manrique}@uniandes.edu.co

Abstract

We present an agent-based system for the TSAR 2025 Shared Task on Readability-Controlled Text Simplification, which requires simplifying English paragraphs from B2+ levels to target A2 or B1 levels while preserving meaning. Our approach employs specialized agents for keyword extraction, text generation, and evaluation, coordinated through an iterative refinement loop. The system integrates a CEFR vocabulary classifier, pretrained evaluation models, and few-shot learning from trial data. Through iterative feedback between the evaluator and writer agents, our system automatically refines outputs until they meet both readability and semantic preservation constraints. This architecture achieved 4th position among participating teams, showing the effectiveness of combining specialized LLMs with automated quality control strategies for text simplification.

1 Introduction

Text simplification is a central task in natural language processing (NLP), aiming to make written content more accessible by reducing linguistic complexity while preserving meaning and fluency. In recent years, *readability-controlled simplification* has gained relevance, where the simplified output must conform to explicit proficiency levels defined by the Common European Framework of Reference for Languages (CEFR). Such control is important for applications in education, second-language learning, and inclusive communication, with recent work showing that instruction-tuned models can follow explicit readability targets (Tran et al., 2024).

The TSAR 2025 Shared Task on Readability-Controlled Text Simplification (Alva-Manchego et al., 2025), organized as part of the EMNLP conference, challenges participants to simplify English paragraphs originally written at the B2 level or above into target levels A2 or B1. Official evaluation metrics include (i) compliance with the target CEFR level, (ii) preservation of the original meaning, and (iii) similarity to human-written reference simplifications. While no training data is provided, organizers released 20 trial examples per target level for format familiarization, making this essentially a few-shot learning challenge that encourages creative combinations of linguistic resources, pretrained models, and novel system architectures.

For this shared task, we developed an agent-based pipeline that leverages the modular nature of the simplification task. Our approach integrates specialized agents for **keyword extraction**, **text generation**, and **evaluation**, orchestrated through an **iterative refinement loop**. The system incorporates diverse tools including a CEFR vocabulary classifier that maps words to proficiency levels (A1-C1), pretrained evaluation models (CEFR classifiers, MeaningBERT, BERTScore), and few-shot examples from the trial data. This iterative feedback mechanism reduces the need for human annotation while ensuring outputs meet both readability and meaning preservation requirements. Our system achieved competitive performance, ranking among the top-5 teams in the shared task. All system components, prompts, and implementation details are publicly available.[1]

In this paper, we describe our system for the TSAR 2025 shared task. Section 2 reviews related work, Section 3 presents our pipeline in detail, Section 4 reports results and analysis, and Section 5 concludes with key insights and directions for future work.

2 Related Work

Readability-controlled generation has recently advanced through instruction tuning that targets fine-

[1]https://github.com/feliperussi/team-uniandes-tsar-2025-shared-task

Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025), pages 211–216

grained complexity levels, showing strong adherence to requested readability and competitive quality (Tran et al., 2024). For longer inputs, multi-agent frameworks coordinate specialized roles to improve document-level coherence and thoroughness, offering an alternative to single-pass simplification (Fang et al., 2025; Lyu and Pergola, 2024). Still, preserving meaning is a key challenge. Paragraph-level human evaluation with reading-comprehension questions shows that even strong systems leave some questions unanswered, highlighting the need for direct meaning checks (Agrawal and Carpuat, 2024).

3 Methodology

3.1 Task Formulation

Given a source paragraph t_0 (B2+ English) and a target CEFR level $\ell^* \in \{\text{A2}, \text{B1}\}$, the goal is to produce a final simplification t^* that: (i) complies with ℓ^* as verified by CEFR classifiers, (ii) preserves semantic content through high similarity scores with t_0, and (iii) maintains fluent, coherent paragraph structure. We enforce meaning preservation through automatic metrics and iterative refinement, using t_n to denote intermediate candidates.

3.2 System Overview

Our system employs an *agent-based* pipeline with iterative refinement to balance readability and meaning preservation (Figure 1). The pipeline processes B2+ source paragraphs through keyword extraction, vocabulary classification, and iterative refinement between Writer and Evaluator agents—whose prompts we co-developed via a hybrid process combining human prompt engineering with LLM-assisted drafting using Gemini 2.5 Pro and GPT-5—until CEFR compliance and semantic similarity thresholds are met. This design ensures outputs meet strict quality requirements through automated validation at each step (detailed components below, thresholds and hyperparameters in Section 4.1).

3.2.1 Keyword Extractor

This agent identifies topic-specific terms that are too complex for the target level ℓ^* and require explicit definition. Given source t_0 and target level $\ell^* \in \{\text{A2}, \text{B1}\}$, it outputs a set $K(t_0, \ell^*) = K$ of keywords that must be defined (not replaced) to preserve meaning. We introduced this separate component because: (i) trial data analysis revealed that certain domain terms must be defined rather than substituted to maintain accuracy, and (ii) LLMs performing end-to-end simplification tend to either over-define common words or miss crucial technical terms.

The agent applies a two-step test to each noun identified by the agent n: (i) can it be replaced by a simple phrase without meaning loss? (ii) would replacement cause awkward repetition (e.g., "gravity" $\rightarrow$ "the force that pulls things down")? Terms failing either test are marked for definition. The agent returns at most two keywords to avoid over-constraining generation, outputting them as JSON: `{"keywords": ["word1", "word2"]}`. The set may be empty if all terms suit level ℓ^*.

3.2.2 Vocabulary Classification

Using the vocabulary lists from Cambridge University Press and Assessment (2025c,a,b) and ESL Lounge (2025), we built a CEFR vocabulary classifier. Given an input text, it returns a dictionary $D(\text{text})$ mapping each word to its CEFR level (A1, A2, B1, B2, C1). We introduced this component because early experiments showed that LLMs without explicit vocabulary guidance tend to either over-simplify text or miss higher-CEFR terms that require replacement. Table 1 reports the size of each CEFR-specific list used during conditioning and evaluation. Regarding words that are out of the scope of $D(\text{text})$, the proposed prompts leave some margin to the models for adding vocabulary.

CEFR Level	Vocabulary Count
A1	1,282
A2	1,228
B1	1,618
B2	595
C1	1,239
Total	**5,962**

Table 1: Vocabulary distribution across CEFR levels

3.2.3 Writer Agent

The writer takes the original B2+ paragraph t_0, target level ℓ^*, keywords $K(t_0, \ell^*)$, vocabulary dictionary D_n, style examples S_{ℓ^*} (that consist in a set of texts in ℓ^*, specifically those known as "reference" in the task trial data), and (for refinement) previous output t_n with feedback f_n. Let $\theta = (t_0, \ell^*, K, S_{\ell^*})$ denote the fixed inputs and $D(t_n)$ the dictionary for each candidate text t_n.

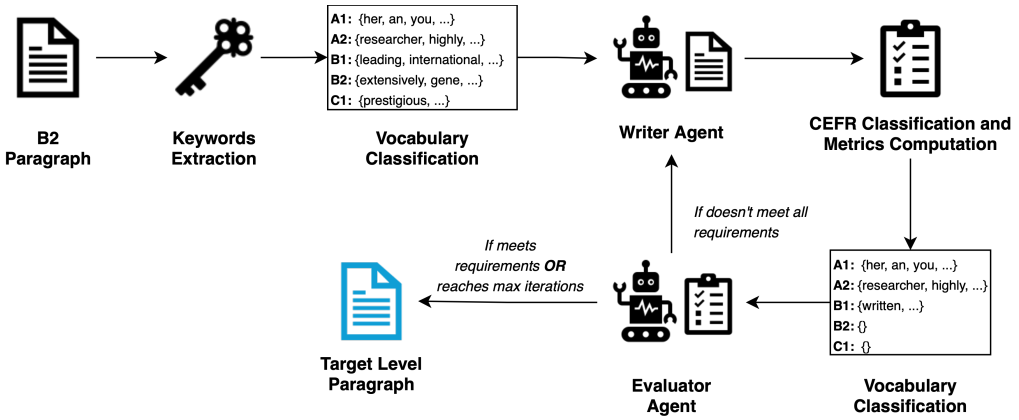

Figure 1: Agent-based pipeline architecture. The system processes B2+ source paragraphs through keyword extraction and vocabulary classification, followed by iterative refinement between the Writer and Evaluator agents. The loop continues until CEFR compliance and semantic similarity thresholds are met or maximum iterations are reached.

Then:

$$t_1 = \text{Writer}(\theta, D(t_0)) \quad (1)$$

$$t_{n+1} = \text{Writer}(\theta, D(t_n), t_n, f_n) \quad \text{for } n \geq 1 \quad (2)$$

In initial generation, the agent: (i) analyzes S_{ℓ^*} to internalize target style from trial data examples, leveraging this data to match expected output patterns while limiting samples to avoid confusion, (ii) extracts core message from t_0, (iii) defines keywords $k \in K$ and uses D to identify vocabulary above target ℓ^*, (iv) restructures sentences to match ℓ^*'s complexity, and (v) returns the simplified paragraph. During refinement iterations, it minimally edits t_n to address issues in f_n. Table 2 summarizes the level-specific constraints.

3.2.4 Automatic Metrics

For each source–hypothesis pair (t_0, t_n) with target level $\ell^* \in \{\text{A2}, \text{B1}\}$, we compute three values:

CEFR label ($\hat{\ell}$). We use three ModernBERT classifiers fine-tuned on different subsets of the UniversalCEFR dataset (Imperial et al., 2025): (i) document-level English, (ii) sentence-level English, and (iii) multilingual texts. Each model $m \in \mathcal{E}$ outputs a predicted level with confidence score. We select the prediction from the model with highest confidence:

$$\ell(t_n) = \text{label}\left(\arg\max_{m \in \mathcal{E}} \text{score}_m(t_n)\right)$$

Following Barayan et al. (2025), we evaluate CEFR compliance using adjacent accuracy (accepting predictions within one level of target) and RMSE between predicted and target level indices.

BERTScore (semantic similarity). We report mean F1 BERTScore (Zhang et al., 2020) between prediction and source:

$$\text{BS}(t_n, t_0) \in [0, 1],$$

interpreted as embedding-based semantic similarity.

MeaningBERT (meaning preservation). We use MeaningBERT (Beauchemin et al., 2023) to assess meaning preservation between t_n and t_0. The metric outputs a percentage score which we normalize to [0,1]:

$$\text{MB}(t_n, t_0) = \frac{\text{MeaningBERT}(t_n, t_0)}{100} \in [0, 1]$$

3.2.5 Evaluator Agent

Given (θ, t_n, D_n) where $D_n = D(t_n)$ is the CEFR vocabulary for the text t_n and automatic metrics (CEFR label $\ell(t_n) = \ell_n$, $\text{BS}_n = \text{BS}(t_n, t_0)$, $\text{MB}_n = \text{MB}(t_n, t_0)$), the evaluator produces feedback:

$$f_n = \text{Eval}(\theta, t_n, D_n, \ell_n, \text{BS}_n, \text{MB}_n) \quad (3)$$

The evaluator assesses outputs through a priority-ordered evaluation pipeline:

1. **Factual integrity:** Verifies numbers, names, dates, locations, and core relations match t_0
2. **Meaning preservation:** Enforces thresholds specified in Section 4.1
3. **CEFR compliance:** Ensures predicted level $\ell_n = \ell^*$

	A2 (elementary)	**B1** (intermediate)
Sentence length	≤12 words, one idea	15–25 words, combined ideas
Connectors	*and, but, because, so, …*	A2 + *moreover, although, however, …*
Keyword handling	define before first use	inline definition
Grammar	simple tenses only	simple + controlled B1 forms
Content scope	essential facts only	preserve key details

Table 2: Level-specific writer constraints controlled by $(K(t_0, \ell^*), D(t_0), S_{\ell^*})$. Aligned with CEFR reading comprehension descriptors (Companion Volume, Ch. 3.1.1.3) (Council of Europe, 2020).

4. **Qualitative audit:** Checks level-specific constraints (see Table 2)

Failure at any priority triggers targeted feedback. The evaluator outputs f_n as JSON containing approval status (`PASS`/`FAIL`), grade (1–10), and explanation of issues with suggested changes. Crucially, the evaluator interprets the automatic metrics (BS_n, MB_n, ℓ_n) and explains their implications to the writer, enabling targeted corrections—a design choice motivated by early experiments where texts had incorrect CEFR classifications or low semantic similarity. If f_n indicates failure, the writer produces t_{n+1} addressing the feedback; if f_n indicates pass, the loop terminates with $t^* = t_n$.

3.2.6 Refinement Loop Rules

We run a Writer–Evaluator loop capped at N maximum iterations. At each iteration n, the candidate t_n is evaluated against the thresholds in Section 4.1; if it passes, the loop terminates and we output t_n. Every iteration logs the candidate and metrics $(t_n,\ \ell_n,\ \text{MB}_n,\ \text{BS}_n,\ f_n)$. If no candidate passes within N iterations, we select the candidate with the highest MB among those correctly classified at the target CEFR level, or if none exist, the candidate with the highest MB overall. Algorithm 1 formalizes this process.

Our repository includes the complete prompts for the Writer, Evaluator, and Keyword Extractor agents; the n8n workflow JSON configurations for agent orchestration; the API implementation with CEFR vocabulary tools; and the trial data preprocessing scripts.

Algorithm 1 Writer-Evaluator Iterative Refinement

Require: Source text t_0 (B2+), target CEFR level $\ell^* \in \{\text{A2}, \text{B1}\}$, max iterations N
Ensure: Simplified text t^*

$K := K(t_0, \ell^*)$, $S := S_{\ell^*}$, $D_0 := D(t_0)$
$\theta := (t_0, \ell^*, K, S)$
$t_1 := \text{Writer}(\theta, D_0)$
for $n = 1$ to N **do**
  $\ell_n := \ell(t_n)$
  $\text{MB}_n := \text{MB}(t_n, t_0)$
  $\text{BS}_n := \text{BS}(t_n, t_0)$
  $D_n := D(t_n)$
  $f_n := \text{Eval}(\theta, t_n, D_n, \ell_n, \text{BS}_n, \text{MB}_n)$
  if f_n indicates PASS **then**
    return $t^* := t_n$
  end if
  if $n < N$ **then**
    $t_{n+1} := \text{Writer}(\theta, D_n, t_n, f_n)$
  end if
end for
$I^* := \{i \in \{1, \ldots, N\} : \ell_i = \ell^*\}$
if $I^* \neq \emptyset$ **then**
  return $t^* := t_j$ where $j := \arg\max_{i \in I^*} \text{MB}_i$
else
  return $t^* := t_j$ where $j := \arg\max_{i=1,\ldots,N} \text{MB}_i$
end if

4 Evaluation and Results

4.1 System Configuration

Our system uses $|S_{\ell^*}| = 10$ style examples per level from TSAR trial data, evaluation thresholds of $\text{MB} \geq 0.75$ and $\text{BS} \geq 0.90$, maximum iterations $N \in \{5, 10\}$ to avoid longer loops, keyword limit $|K| \leq 2$ irreplaceable terms per paragraph, and temperature 0 for all models to ensure deterministic outputs. These values were selected based on trial data analysis to balance quality and efficiency. Check Table 3 for the summary of each run.

Run	**Writer**	**Evaluator**	N
Run 1	Gemini 2.5 Pro	Gemini 2.5 Pro	5
Run 2	Gemini 2.5 Flash	Gemini 2.5 Flash	10
Run 3	GPT-OSS-120B	Gemini 2.5 Pro	10

Table 3: System configurations evaluated. Run 1 uses Gemini 2.5 Pro (Google AI for Developers, 2025), Run 3 uses GPT-OSS-120B (Together AI, 2025; OpenAI, 2025).

We orchestrate the agents with n8n (2025) and a REST API serving level-specific control inputs. The trial data provided style examples and vocabulary guidance.

4.2 Results

Our team Uniandes achieved 4th place in the TSAR 2025 Shared Task, with the resulting metrics shown in Table 4. The Gemini 2.5 Flash configuration provided the best balance between CEFR compliance and efficiency, while the hybrid GPT-OSS-120B/Gemini 2.5 Pro configuration excelled at meaning preservation, as seen in Table 5. All systems maintained high semantic similarity (BS > 0.92) while successfully adapting texts to target CEFR levels.

Run	F1	Adj. Acc.	RMSE
Run 1	0.972	1.00	0.212
Run 2	**0.985**	**1.00**	**0.200**
Run 3	<u>0.851</u>	<u>0.97</u>	<u>0.510</u>

Table 4: CEFR Compliance metrics on TSAR 2025 test set. Best scores in bold and worst in underline.

Run	MB-Orig	MB-Ref	BS-Orig	BS-Ref
Run 1	<u>0.817</u>	**0.814**	**0.936**	**0.934**
Run 2	0.823	<u>0.803</u>	0.934	0.930
Run 3	**0.847**	0.813	0.933	0.928

Table 5: Comparison between meaning preservation metrics on TSAR 2025 test set. Best scores in bold and worst in underline. MB and BS stand for MeaningBERT and BERTScore respectively.

5 Discussion

Our experiments showed distinct trade-offs across model configurations. While Run 2 achieved the best CEFR compliance and Run 1 demonstrated high precision and provided the most reliable balance between CEFR compliance and meaning preservation. The hybrid Run 3 struggled with level targeting. For meaning preservation, Run 1 is more stable and shows less overfitting; Run 3, despite having a higher MB-Orig, has a lower MB-Ref, indicating overfitting to the source text.

After analyzing simpler agent configurations on trial data, several design improvements significantly increased performance. We found that using all 20 trial examples was counterproductive and wasteful of tokens, so limiting to 10 samples optimized both performance and efficiency. Style examples enhanced output consistency, keyword extraction preserved domain-specific meaning through definitions, and CEFR vocabulary classification prevented inconsistent term replacements. However, some configurations required many iterations to achieve correct CEFR levels, suggesting convergence problems. Our MeaningBERT threshold of 0.75 may have been conservative—higher thresholds could enforce stronger semantic preservation.

Furthermore, Run 3 demonstrated that hybrid architectures with open models as writers can excel at semantic preservation despite weaker level control. This suggests that open models could be valuable as writer agents when paired with strong closed-model evaluators like Gemini 2.5 Pro, potentially offering cost-effective alternatives to fully proprietary systems.

6 Limitations

Key limitations include expensive and unpredictable token generation from iterative refinement, with some texts requiring many iterations and extended processing times. The CEFR vocabulary coverage could be potentially missing some terms. The token consumption was highly variable across different texts, making cost prediction difficult.

Our CEFR and semantic-similarity thresholds were intentionally conservative to favor coverage and reduce non-convergence. More ambitious (stricter) thresholds might yield higher precision in meaning preservation and level control, but at the cost of lower acceptance rates and longer refinement. Exploring adaptive or curriculum-style thresholds is left for future work.

7 Lay Summary

This work builds a system that turns hard-to-read English paragraphs into easier ones for learners, at two target levels: A2 (elementary) and B1 (intermediate). The main goal is to make the text simpler without changing its meaning. To do this, the system uses three "agents" that work together in a loop: one finds important, difficult words that should be defined (not replaced), another writes a simpler version, and a third checks the result. If the check finds problems—like lost meaning or the level being too hard—the writer tries again. This repeat-and-improve cycle continues until the text is both simple enough and faithful to the original.

To guide the writing, the system uses word lists tied to the CEFR levels (A1–C1) and short example texts that show the expected style. To check quality automatically, it uses tools that (1) estimate the reading level, and (2) measure how closely the new

text matches the original meaning. Several model setups were tested and set clear passing rules before accepting any output.

In a public competition on readability-controlled simplification, this approach ranked among the top teams (4th place). The results show that combining specialized roles with automatic checks can reliably simplify text while keeping its meaning. As limitations, some paragraphs need many rounds (which can be costly), and the vocabulary lists may not cover every word.

References

Sweta Agrawal and Marine Carpuat. 2024. Do text simplification systems preserve meaning? a human evaluation via reading comprehension. *Transactions of the Association for Computational Linguistics*, 12:432–448.

Fernando Alva-Manchego, Regina Stodden, Joseph Marvin Imperial, Abdullah Barayan, Kai North, and Harish Tayyar Madabushi. 2025. Findings of the TSAR 2025 Shared Task on Readability-Controlled Text Simplification. In *Proceedings of the Fourth Workshop on Text Simplification, Accessibility, and Readability (TSAR 2025)*, Suzhou, China. Association for Computational Linguistics. Latest update: September 2025; archived at https://web.archive.org/web/20250914180215/https://tsar-workshop.github.io/shared-task/.

Abdullah Barayan, Jose Camacho-Collados, and Fernando Alva-Manchego. 2025. Analysing zero-shot readability-controlled sentence simplification. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6762–6781, Abu Dhabi, UAE. Association for Computational Linguistics.

David Beauchemin, Horacio Saggion, and Richard Khoury. 2023. MeaningBERT: assessing meaning preservation between sentences. *Frontiers in Artificial Intelligence*, 6. Publisher: Frontiers.

Cambridge University Press and Assessment. 2025a. Vocabulary List for A2 level.

Cambridge University Press and Assessment. 2025b. Vocabulary List for B1 level.

Cambridge University Press and Assessment. 2025c. Vocabulary List for Pre A1 Starters, A1 Movers and A2 Flyers.

Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, teaching, assessment — Companion volume*. Council of Europe Publishing, Strasbourg. Reading comprehension descriptors (see Chapter 3.1.1.3).

ESL Lounge. 2025. ESL Lounge: Learn English with ESL Lounge.

Dengzhao Fang, Jipeng Qiang, Xiaoye Ouyang, Yi Zhu, Yunhao Yuan, and Yun Li. 2025. Collaborative document simplification using multi-agent systems. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 897–912, Abu Dhabi, UAE. Association for Computational Linguistics.

Google AI for Developers. 2025. Gemini models: Gemini API | Google AI for Developers. Latest update: August 2025; archived at https://web.archive.org/web/20250825013047/https://ai.google.dev/gemini-api/docs/models#previous-experimental-models.

Joseph Marvin Imperial, Abdullah Barayan, Regina Stodden, Rodrigo Wilkens, Ricardo Munoz Sanchez, Lingyun Gao, Melissa Torgbi, Dawn Knight, Gail Forey, Reka R. Jablonkai, Ekaterina Kochmar, Robert Reynolds, Eugenio Ribeiro, Horacio Saggion, Elena Volodina, Sowmya Vajjala, Thomas Francois, Fernando Alva-Manchego, and Harish Tayyar Madabushi. 2025. Universalcefr: Enabling open multilingual research on language proficiency assessment. *Preprint*, arXiv:2506.01419.

Chen Lyu and Gabriele Pergola. 2024. Society of Medical Simplifiers. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 61–68, Miami, Florida, USA. Association for Computational Linguistics.

n8n. 2025. n8n: Workflow automation platform (v1.109.0). https://github.com/n8n-io/n8n/releases/tag/n8n@1.108.0. Version 1.109.0, released August 25, 2025.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Together AI. 2025. gpt-oss-120B API. Archived at https://web.archive.org/web/20250821140846/https://www.together.ai/models/gpt-oss-120b.

Hieu Tran, Zonghai Yao, Lingxi Li, and Hong Yu. 2024. Readctrl: Personalizing text generation with readability-controlled instruction learning. *Preprint*, arXiv:2406.09205.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.