NIL-UCM at PROFE 2026: Prompting, Fine-Tuning and Multi-Agent Strategies for Reading Comprehension

Pablo Folgueira[1], Sandra Conde[1] and Alberto Díaz[2]

[1]Facultad de Informática, Universidad Complutense de Madrid, Spain
[2]Facultad de Informática and ITC, Universidad Complutense de Madrid, Spain

Abstract

This paper evaluates Large Language Models (LLMs) and multi-agent architectures for reading comprehension tasks within the PROFE 2026 framework. Comparing optimized open source models, such as Qwen 3.5 9B and Gemma 4 E4B, to a multi-agent system powered by Gemini 2.5 Flash Lite, our experiments show that while the multi-agent setup achieved the highest overall accuracy, baseline models delivered highly competitive zero-shot results. The evaluation highlights critical trade-offs by demonstrating that complex iterative systems introduce significant latency and carbon emissions for minimal accuracy gains. Ultimately, we demonstrate that smaller models can efficiently handle complex language assessments, suggesting dynamic routing to balance performance with sustainable deployment in educational technology.

Keywords

multiple choice, matching, Zero-Shot Prompting, Chain-of-Thought, Fine-Tuning, Spanish reading comprehension, LLMs, Multi-Agent, computational efficiency, environmental impact

1. Introduction

The rapid advancement of Large Language Models (LLMs) has significantly transformed the field of Natural Language Processing (NLP), enabling systems to tackle increasingly complex tasks that require deep language understanding, logical inference, and contextual reasoning. Despite these remarkable capabilities, evaluating such systems under realistic and controlled conditions remains a fundamental challenge. Assessments that rely on task-specific training data risk overestimating model performance, as systems may leverage dataset patterns rather than demonstrating genuine reasoning and generalization skills.

In this context, automated language proficiency assessment emerges as a particularly relevant application area. As AI tools become more integrated into educational settings, understanding how well current models perform when placed under the same conditions as human learners, with no task-specific preparation and a wide variety of exercise types, becomes increasingly important. The PROFE 2026 shared task [1] addresses this challenge directly, using official Spanish proficiency examinations from the Instituto Cervantes that span all proficiency levels from A1 to C2. Moreover, this edition introduces a new multimodal component in which some exercises include images that may or may not be relevant to solving the task, adding a significant layer of complexity to the evaluation.

In this paper, we explore prompting strategies, fine-tuning, and multi-agent architectures applied to different LLMs for the multiple-choice and matching subtasks of PROFE 2026. Our experiments evaluate the accuracy of each approach while also examining their computational cost and environmental impact, with the aim of identifying solutions that balance performance with responsible and sustainable deployment.

IberLEF 2026, September 2026, León, Spain
pablofol@ucm.es (P. Folgueira); saconde@ucm.es (S. Conde); adiazest@ucm.es (A. Díaz)
0009-0005-3725-3954 (P. Folgueira); 0009-0009-3535-1330 (S. Conde); 0000-0003-1966-3421 (A. Díaz)

2. Task Description

The PROFE 2026 shared task [1], held as part of the IberLEF 2026 [2], aims to evaluate the Spanish language proficiency of automated systems under the same conditions applied to human learners. To achieve this, the task provides no specific training material, trying to test the true generalization and reasoning capabilities of the evaluated systems without relying on task-specific specialization.

A significant novelty in this edition of the task is the shift beyond text-only comprehension to encompass multimodal reasoning, as exercises can now include images. These visual elements vary in their utility, as some provide essential cues or serve as candidate answers while others act as irrelevant distractors. Consequently, the necessity to process multimodal inputs and discern the relevance of visual information introduces a significant new challenge for the solutions proposed for the task.

2.1. Dataset

The evaluation is based on the IC-UNED-RC-ES dataset [3], which consists of real exams created by human experts at the Instituto Cervantes to assess Spanish language proficiency. These exams have been digitalized to properly adapt them for this evaluation task, creating a dataset that covers all language proficiency levels from A1 to C2.

For this edition, the task uses the remaining 50% of the dataset that was not evaluated in the previous year, now including the new image-based exercises. Finally, to rigorously evaluate generalization and prevent data contamination or overfitting in LLMs, the gold standard annotations are strictly withheld by the organizers.

2.2. Addressed Subtasks

While PROFE 2026 proposes three subtasks corresponding to different types of exercises, our participation and the scope of this paper focus exclusively on the following two:

- **Multiple choice subtask:** In this subtask, systems are provided with a reference text and a set of related questions. For each question, systems must select the single correct answer from a given list of candidates. Furthermore, due to the inclusion of multimodal elements in this edition, some questions present images instead of text as candidate answers. In these cases, the system must choose the image that best answers the question.
- **Matching subtask:** In this subtask, each exercise contains two sets of texts. Systems must find the text in the second set that best matches each element in the first set. There is only one possible matching per text, but the first set may contain extra, unnecessary texts that act as distractors. Additionally, texts may include an associated image, which can either provide helpful visual context to solve the exercise or simply serve as a distractor.

Regarding the evaluation of these subtasks, the metric used for both of them is accuracy. For the Multiple-Choice subtask, it is measured as the proportion of correctly answered questions, while for the Matching subtask, it is calculated as the proportion of correctly matched texts. This straightforward metric is used to ensure a direct and fair comparison between the performance of the automated systems and human examinees under the exact same conditions.

3. Methodology

In this section, we describe the proposed approaches for solving the PROFE 2026 subtasks in which we participated. Our methodology is primarily driven by the use of open-source Large Language Models (LLMs) and explores various prompt engineering, fine-tuning and multi-agent strategies.

3.1. Multiple-Choice Subtask

For the multiple-choice subtask, we focused on evaluating different multimodal LLMs. In total, we designed and tested five distinct approaches to determine the most effective strategy for reading comprehension.

- **Zero-Shot Prompting:** We evaluated Qwen-3.5-9B [4] and Gemma-4-E4B-it [5] in a zero-shot setting, with a system prompt that assigned them the role of an expert teacher in Spanish reading comprehension. Clear instructions were provided to base the answer exclusively on the provided text, strictly avoiding the use of external knowledge. To facilitate automatic evaluation, the models were constrained to output only the letter corresponding to the correct option, without any introductory text or explanation.
- **Multi-Agent Ensemble with LLM-as-a-Judge:** We implemented an ensemble system using three different models: Qwen-3.5-9B, Gemma-4-E4B-it, and Ministral-3-8B-Instruct [6]. Initially, all three models generated an independent response using the exact same zero-shot prompting strategy and system instructions described in the previous approach. Once all responses were collected, if the models reached a unanimous agreement, that answer was selected as the final decision. In cases of discrepancy, Gemma-4-E4B-it was employed as an impartial "Judge". This model was provided with all the available options, along with specific information about which answers were chosen by the ensemble and their respective number of votes, and was instructed to critically analyze the alternatives based solely on the source text. Finally, we asked the model to reason step by step to discard the incorrect options, and then output only the letter of the correct answer.
- **Chain-of-Thought (CoT) Prompting:** To investigate whether explicit reasoning could improve performance in this task, we applied a CoT strategy to Gemma-4-E4B-it. The prompt instructed the model to act as an expert and analyze the text globally (paying attention to discourse markers, verb tenses, and pronouns). Specifically, we asked the model to evaluate all the available options and discard the incorrect ones one by one, providing clear arguments based on the text while ignoring option length. To ensure structured parsing, the model was forced to output a valid JSON object containing two fields: `reasoning` (the step-by-step explanation) and `answer` (the final selected option).
- **Task-Specific Fine-Tuning:** Finally, we fine-tuned the Gemma-4-E4B-it model using the ReCoRES dataset [7] to adapt its weights to this specific reading comprehension task. This dataset is built from Peruvian university entrance examinations and comprises 439 texts and 1,822 questions that require advanced reasoning skills. A notable feature of ReCoRES is that questions have a variable number of candidate answers (ranging from 2 to 7) and include short explanations justifying the correct choice. However, during our preprocessing phase, we excluded these explanatory rationales, training the model to output solely the correct answer choice. Furthermore, we deliberately maintained the variance in the number of candidate options rather than normalizing them to prevent the model from overfitting to a fixed number of answers, thereby improving its flexibility and robustness when facing the unpredictable lengths of the PROFE 2026 exercises.

3.2. Matching Subtask

To address the specific challenges of the matching subtask, we initially evaluated several open-source models in a zero-shot setting to establish a reference baseline. Subsequently, we developed a more complex multi-agent system in which different LLMs collaborated to reach the final solution.

3.2.1. Zero-Shot Strategies

To guide the models in our zero-shot approach, we designed two system prompts based on the specific characteristics of each type of exercise. Given that the required response format for this subtask was

more complex than in the previous one, we incorporated an example of the expected JSON output directly into these instructions. Once these templates were defined, the first step in our pipeline consisted of dynamically identifying the type of exercise based on the number of questions and candidate answers, enabling the system to automatically apply the appropriate prompt.

- **Case 1 (Repeatable Answers):** This prompt is applied when there are more questions than answers. It instructs the model that the options from the first set (answers) can be mapped to multiple elements in the second set (questions).
- **Case 2 (Presence of Distractors):** This one is used when there are more answers than questions. The prompt explicitly warns the model about distractors that must be ignored, enforcing a strict one-to-one mapping for valid options.

In both cases, the models were constrained to format their final output as a valid JSON dictionary mapping element IDs to option IDs, omitting any introductory text or reasoning. We evaluated this zero-shot baseline strategy using Qwen-3.5-9B and Gemma-4-E4B-it.

3.2.2. Multi-Agent Collaborative Architecture

Our objective with this approach was to investigate whether orchestrating a collaborative framework of LLMs, where each agent is specialized in a concrete subtask, could significantly enhance overall performance. This solution prevents a single model from having to manage all complex reasoning tasks independently. To this end, we developed a multi-agent system where all agents use the Gemini 2.5 Flash Lite [8] model. Within this architecture, each agent is assigned a distinct role tailored to a concrete task to solve the exercise. The agents involved in this architecture are explained below:

1. The Orchestrator Agent: Acting as the initial router, this agent analyzes the exercise instructions and the number of questions and candidate answers in the given sets to determine the execution path: **Route A** for "many-to-few" exercises (where answers can be repeated), or **Route B** for "one-to-one" exercises (which contain candidate answers acting as distractors).

2. Route A (Proposer-Critic Collaboration): When the Orchestrator Agent determines that candidate answers can be reused across questions, the system handles the matching task through a cooperative Proposer-Critic process:

- **Resolver Agent (Proposer):** Analyzes a single question against all available answers to propose a candidate match, adjusting its reasoning based on contextual feedback whenever provided by the reviewer.
- **Reviewer Agent (Critic):** Evaluates the option proposed by the resolver against the rest of the possible answers to prevent the selection of misleading options. If it determines that the proposed option is incorrect, it provides constructive feedback, prompting the Resolver to refine its choice until a final answer is selected.

This iterative loop can repeat up to three times per question, but if no agreement is reached after the third attempt, the system accepts the Resolver's last proposed answer before moving on to the next question.

3. Route B (Bidirectional Consensus and Auditing): On the other hand, for exercises that require strict one-to-one mappings and include distractors, the system processes the entire set simultaneously to find a solution through a Bidirectional Consensus and Auditing approach:

- **Direct Agent (Questions to Answers):** Processes the questions sequentially, mapping each one to the best fitting text from the first set. Once a text is selected, it is removed from the available options for the following questions. This progressive elimination shrinks the search space for the remaining questions, resulting in an initial draft of one-to-one mappings.

- **Inverse Agent (Answers to Questions):** To avoid relying solely on the first agent's proposal, this model performs the process in reverse. It evaluates each text from the first set one by one, associating it with the question that it considers the best fit or leaving it unassigned if it is a distractor, thereby generating an independent, alternative set of mappings.

Once the Direct and Inverse Agents have generated their drafts, the system uses cross-validation to establish the final solution. For each question, it does so by checking the following scenarios:

- **Perfect Agreement:** If the direct and inverse predictions match, the system accepts the pair.
- **Single-Sided Claim:** If the Inverse Agent does not assign an answer, but the Direct Agent does, the system accepts the Direct Agent's choice.
- **Conflict Identification:** If there is a mismatch between the two drafts, the system aggregates all possible options into a candidate pool, which includes the answer selected by the Direct Agent alongside any answers that claimed the question during the inverse phase.

To resolve these discrepancies, the system relies on a final specialized agent:

- **Auditor Agent (Conflict Resolution):** In scenarios involving conflicting mappings or multiple candidate texts, this agent acts as a definitive tie-breaker. Given the target question and its candidate pool, it reviews each option one by one to determine the best fit and select the final answer.

To ensure that no question is left unanswered, a final sweeper phase is executed after conflict resolution. The system prompts the Direct Agent to match any remaining orphaned questions exclusively with the leftover pool of answers. If the model cannot find a valid match, a fallback mechanism simply assigns one of the leftover texts to ensure all questions are answered.

System Workflow (figure 1): Execution begins with the Orchestrator Agent determining the appropriate path. If Route A is selected, the system resolves questions sequentially through an iterative feedback loop. If Route B is chosen, the Direct and Inverse Agents generate independent drafts that are cross-validated to establish consensus, while any conflicting pairs are resolved by the Auditor Agent. To guarantee completeness, a final sweeper phase matches any unassigned questions with the leftover answers. Ultimately, this multi-agent process ensures a fully resolved and accurate set of mappings.

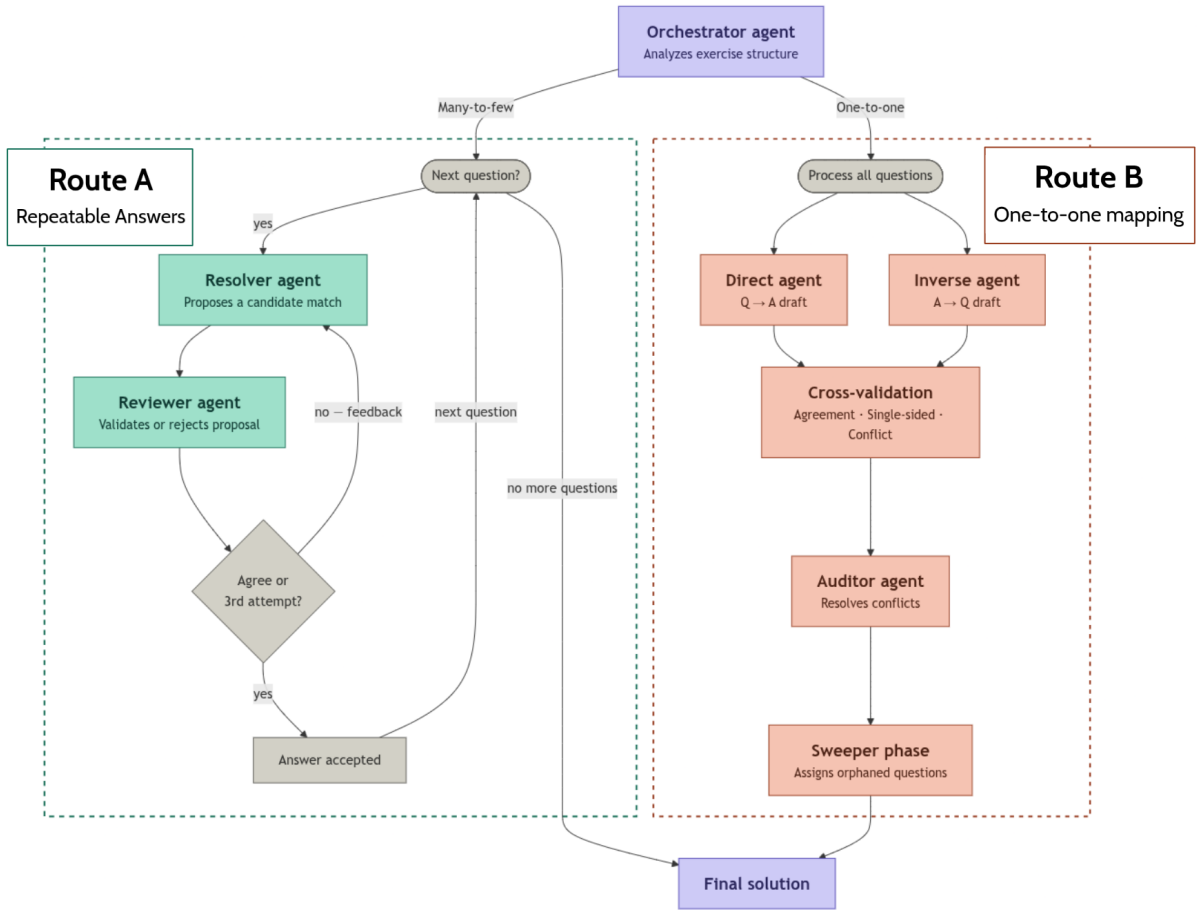

Figure 1: Architecture of the proposed multi-agent system for the matching task.

4. Results and Discussion

This section presents the evaluation results for both multiple-choice and matching tasks. Table 1 summarizes the performance of the solutions we tested, which are discussed below.

Table 1
Performance comparison for the evaluated solutions across multiple-choice and matching tasks.

Task Type	Model / Approach	Accuracy (%)
Multiple Choice	Multi-Agent Ensemble	**90.55**
	Qwen 3.5 9B (Zero-shot)	88.85
	Gemma 4 E4B (Chain of Thought)	88.42
	Gemma 4 E4B (Zero-shot)	85.63
	Gemma 4 E4B (Fine-tuned)	85.46
Matching	Multi-Agent System	**86.51**
	Qwen 3.5 9B (Zero-shot)	86.11
	Gemma 4 E4B (Zero-shot)	83.57

4.1. Multiple Choice

In the multiple choice task, the proposed multi-agent ensemble achieved the highest accuracy (90.55%), closely followed by the zero-shot strategy with Qwen 3.5 9B (88.85%). As a model released in early 2026, its performance demonstrates the remarkable baseline capabilities and out-of-the-box reasoning power of LLMs for this type of task.

Gemma 4, another model released this year, did not reach the exact same results but remained quite close, achieving an 85.63% accuracy in a zero-shot setup. Furthermore, using a Chain of Thought (CoT) prompting strategy clearly improved Gemma 4's reasoning, bringing its performance up to the same level as Qwen 3.5 9B. This shows that allowing these models to reason by discarding options one by one and providing clear arguments based on the text can improve the final score. This step-by-step approach helps the model pay attention to distractors or specific text details that it might easily miss if forced to answer directly, which can be useful for the more complex exam levels.

Conversely, the fine-tuning approach applied to Gemma 4 failed to outperform the other evaluated strategies. This counterintuitive outcome may be attributed to two main factors: first, a potential overfitting to the training dataset, which comprised texts more complex than those evaluated in this specific task; and second, the possibility that the fine-tuning process constrained the model's generalized reasoning capabilities, which seem to be crucial for resolving multiple-choice questions.

4.1.1. Environmental Impact and Execution Time

Beyond task performance, we measured the computational cost of the tested approaches in terms of execution time and estimated carbon footprint (kg of CO_2 equivalent) using the CodeCarbon package [9]. These metrics, collected in Google Colab environments equipped with either NVIDIA T4 or A100 GPUs, are summarized in Table 2.

Table 2
Execution time and environmental impact (CO_2eq emissions) for the multiple-choice task across different setups.

Approach	Hardware	Execution Time	Emissions (kg CO_2eq)
Qwen 3.5 9B (Zero-shot)	A100	8.7 min	0.0175
Gemma 4 (Zero-shot)	T4	25.2 min	0.0065
Gemma 4 (Fine-tuned)[a]	A100	21.5 min	0.0367
Gemma 4 (Chain of Thought)	A100	4.59 hours	0.3502
Multi-Agent Ensemble[b]	Mixed (A100 & T4)	∼4.24 hours	0.2546

[a] Includes 15.5 min (0.0263 kg CO_2eq) for fine-tuning and 6 min (0.0104 kg CO_2eq) for inference.
[b] Cumulative cost of Qwen, Gemma, Ministral (T4), and the Gemma 4 Judge agent.

These results highlight a clear trade-off between accuracy and resource efficiency. Although the multi-agent ensemble and the CoT strategy achieved the highest scores, their computational cost is quite high. The ensemble took over 4 hours to run and produced 0.2546 kg of CO_2eq, mostly due to the Gemma 4 Judge agent (3.12 hours and 0.2131 kg of CO_2eq). Likewise, CoT was the most demanding single-model setup, generating 0.3502 kg of CO_2eq in almost 4.6 hours.

On the other hand, zero-shot executions were much more efficient. Qwen 3.5 9B stands out as the most balanced option for this task, nearly achieving the best accuracy in just 8.7 minutes with a very low carbon footprint of 0.0175 kg CO_2eq.

4.2. Matching

For the matching exercises, baseline models demonstrated remarkably strong out-of-the-box capabilities, particularly Qwen 3.5 9B, which is notable given the task's complexity, as it requires evaluating multiple texts simultaneously and correctly mapping them despite distractors. The results suggest that dynamically adapting the prompt based on the exercise type significantly helps the models, as clearly

defining the structural constraints and task objectives allows them to better adhere to instructions and execute the mapping effectively.

Although the proposed multi-agent system achieved the highest accuracy, the difference compared to a simpler solution like a zero-shot Qwen 3.5 9B is very small (86.51% vs. 86.11%). This once again proves the strong capabilities of modern baseline models when solving these types of reading comprehension tasks. Since our multi-agent architecture relies on Gemini 2.5 Flash Lite (a model released in January 2025), integrating newer models could likely improve the ensemble's results. However, it is hard to beat these small, highly capable models by a large margin anyway. For future work, it would be interesting to benchmark the system using more recent state-of-the-art models and explore architectural refinements to see if the performance gains can justify the added complexity of a multi-agent approach.

4.2.1. Execution Time and Emissions for Zero-Shot Strategies

In terms of computational efficiency and environmental impact, the zero-shot strategies presented highly sustainable results for this task. We recorded the execution time and estimated carbon footprint for the standalone zero-shot models running on a Google Colab environment equipped with an NVIDIA A100 GPU, as detailed in Table 3.

Table 3
Execution time and environmental impact (CO_2eq emissions) for zero-shot strategies in the matching task.

Model (Zero-shot)	Execution Time	Emissions (kg CO_2eq)
Qwen 3.5 9B	13.0 min	0.0206
Gemma 4 e4b-it	18.6 min	0.0216

These results show that Qwen 3.5 9B is a highly optimized baseline. It not only gets almost the best accuracy but also runs more efficiently than Gemma 4, finishing the task in less time and with a slightly lower carbon footprint. This highlights how smaller, capable models can handle complex reading comprehension tasks without a heavy environmental cost.

Although we didn't track exact execution metrics for the multi-agent system on the matching task, it clearly took much longer to run in practice. Relying on multiple iterative steps and calling cloud-based models via APIs naturally requires a lot more computing power. Because of this, it is safe to assume that the multi-agent approach leaves a much larger carbon footprint than the fast, local execution of zero-shot models.

5. Conclusions and Future Work

In this work, we explored how both LLMs and multi-agent systems perform when solving reading comprehension exercises. Specifically, we assessed their capabilities across multiple-choice and matching exercises within the PROFE 2026 evaluation framework. Our experiments demonstrate that while the proposed multi-agent systems achieved the highest overall accuracy (90.55% in multiple-choice and 86.51% in matching), small multimodal models like Qwen 3.5 9B performed very well in zero-shot settings. This clearly shows that modern LLMs already have strong reasoning skills, especially when guided by prompts tailored to the specific exercise.

The evaluation also revealed critical trade-offs between accuracy, architectural complexity, and resource efficiency. Strategies such as Chain of Thought (CoT) significantly improved reasoning for models like Gemma 4, but at a high computational and environmental cost. Conversely, fine-tuning Gemma 4 resulted in a slight performance drop, suggesting potential overfitting to complex texts and a restriction of the model's generalized reasoning skills. Furthermore, the minimal accuracy gain provided by the multi-agent system in the matching task compared to its high latency and carbon footprint raises important questions about the scalability of iterative workflows for these types of tasks.

Future work should focus on optimizing multi-agent architectures to reduce unnecessary API calls and mitigate their environmental impact. A practical approach is using a dynamic routing setup by relying on a small model for regular questions and saving the complex multi-agent process just for the hardest edge cases, finding the best balance between performance and efficiency to avoid wasting resources. Additionally, measuring the performance gap between these architectures and small open-source models will help determine the actual value added by complex multi-agent ensembles when solving these tasks.

Ultimately, this research highlights a clear path forward for automated language assessment. But as AI integration grows in educational technology, finding a healthy balance between getting accurate results, keeping costs down, and protecting the environment will be key to doing it responsibly.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini in order to check grammar and spelling, as well as to paraphrase and reword text. After using this tool, the authors reviewed and edited the content as needed, and they take full responsibility for the publication's content.

References

[1] A. Rodrigo, S. Moreno-Álvarez, A. Pérez García-Plaza, A. Peñas, R. Agerri, J. Fruns-Jiménez, I. Soria-Pastor, Overview of profe at iberlef 2026: Language proficiency evaluation, Procesamiento del Lenguaje Natural 77 (2026).

[2] A. Bonet-Jover, J. A. González-Barba, L. Chiruzzo, Overview of iberlef 2026: Natural language processing challenges for spanish and other iberian languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS.org, 2026.

[3] A. Peñas, A. Rodrigo, J. Fruns-Jiménez, I. Soria-Pastor, S. Moreno-Álvarez, A. Pérez García-Plaza, J. Reyes-Montesinos, A spanish language proficiency dataset for ai evaluation, Information 17 (2026) 159. doi:10.3390/info17020159.

[4] Qwen Team, Qwen3.5: Towards native multimodal agents, 2026. URL: https://qwen.ai/blog?id=qwen3.5.

[5] Google DeepMind, Gemma 4, https://deepmind.google/models/gemma/gemma-4/, 2026.

[6] Mistral AI, Ministral 3 8b model card, https://docs.mistral.ai/models/model-cards/ministral-3-8b-25-12, 2025.

[7] M. A. S. C. y Diego Diestra y Rodrigo López y Erasmo Gómez y Arturo Oncevay y Fernando Alva-Manchego, Overview of recores at iberlef 2022: Reading comprehension and reasoning explanation for spanish, Procesamiento del Lenguaje Natural 69 (2022) 281–287. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6448.

[8] Google Cloud, Gemini 2.5 flash-lite documentation, https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/2-5-flash-lite?hl=es-419, 2025.

[9] B. Courty, V. Schmidt, A. Goyal-Y, et al., Codecarbon: Estimate and track carbon emissions from machine learning computing, https://github.com/mlco2/codecarbon, 2021.

Online Resources

The following online resources are related to the work presented in this paper.

Code Repository

- **NIL-UCM PROFE 2026 proposed solutions**: Source code for all approaches described in this paper, including prompting strategies, fine-tuning scripts, and the multi-agent architectures for

both the multiple-choice and matching subtasks.
https://github.com/NILGroup/TFM2526-ExamenesComprensionLectora

Datasets

- **IC-UNED-RC-ES**: https://zenodo.org/records/15790139
- **ReCoRES**: https://portal.odesia.uned.es/en/dataset/recores-2022

Models

- **Qwen3.5-9B**: https://huggingface.co/Qwen/Qwen3.5-9B
- **Gemma-4-E4B-it**: https://huggingface.co/google/gemma-4-E4B-it
- **Ministral-3-8B-Instruct**: https://huggingface.co/mistralai/Ministral-3-8B-Instruct-2512
- **Gemini 2.5 Flash Lite**: https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash-lite

Tools

- **CodeCarbon**: Open-source Python package used to estimate the carbon footprint (kg of CO_2 equivalent) and track the computational cost of all experiments.
https://github.com/mlco2/codecarbon