UCO-UC-CICESE-Plenitas Team at PROFE 2026: Exploring in the Language Proficiency Evaluation Using Ensembles

Yoan Martínez-López[1,2,*], Ireimis de las Mercedes Leguen de Varona[3], Julio Madera[3], Ana Orquidea López Correoso[6], Luis Quintero Domínguez[4], Ansel Rodríguez-González[5], Kristell Aguilar Alarcón[1,2], Carlos de Castro Lozano[1,2] and Jose Miguel Ramirez Uceda[1,2]

[1]*EATCO. Universidad de Córdoba, Córdoba, Spain*
[2]*Plénitas, C/ Le Corbusier s/n, 14005 Córdoba, Spain*
[3]*Universidad de Camaguey, Circunvalación Norte, Camino Viejo Km 5 y 1/2, Camaguey, Cuba*
[4]*Universidad de Sancti Spíritus, Cuba*
[5]*Centro de Investigación Científica y de Educación Superior de Ensenada, Unidad Académica Tepic, México*
[6]*IPVCE "Máximo Gómez Báez", Circunvalación Norte, Camaguey, Cuba*

Abstract

This paper describes the participation of the UCO-UC-CICESE-Plenitas team in the PROFE 2026 shared task on Spanish Language Proficiency Evaluation, co-located with IberLEF 2026. The task is formulated as a zero-shot reading-comprehension problem over authentic examinations developed by the Instituto Cervantes for the certification of Spanish as a foreign language at different CEFR levels, and is structured into three complementary subtasks: Multiple Choice, Matching, and Fill-in-the-Gap. A complete-exam metric (proportion of exams passed at a 0.6 accuracy threshold) is additionally computed for teams participating in all three subtasks, mirroring the passing criterion applied to human candidates. We propose a modular pipeline that combines an instruction-tuned multilingual sentence encoder (intfloat/multilingual-e5-large-instruct) with an open-weight instruction-tuned LLM served through Ollama (gemma4:31b-cloud). The reasoning core is implemented as three subtask-specific solvers that share the same backends: a hybrid LLM then embedding solver for Multiple Choice, in which an embedding-based cosine-similarity fallback recovers from low-confidence LLM predictions; and two embedding-based optimal-assignment solvers for Matching and Fill-in-the-Gap, which formulate the task as a linear assignment problem over a similarity (or local-coherence) matrix and solve it globally with the Hungarian algorithm, thereby enforcing the bijectivity constraint that prompt-based LLMs are known to violate. On the official evaluation, our system was the only participant to submit predictions for the three subtasks simultaneously, achieving the best score on Multiple Choice (93.44) and a final Exam-level score of 81.01, the only such score in the official leaderboard. Internal ablations identify the capacity of the generative model as the single most influential factor in the pipeline, with the embedding model acting as a secondary.

Keywords

LLMs, exam, embeddings, ensembles

1. Introduction

Deep language comprehension is essential for capturing semantic nuances and supporting logical inference in Natural Language Processing (NLP) [1, 2]. A persistent challenge in this domain is the development of new evaluation resources, which typically demands considerable human effort. To mitigate this bottleneck, the community has widely reused existing human reading comprehension datasets such as RACE [3], enabling the benchmarking of automated systems against human-level performance. Nonetheless, the majority of these resources are predominantly available in English and

IberLEF 2026, September 2026, León, Spain
*Corresponding author.
ymlopez2022@gmail.com (Y. Martínez-López); ireimis.leguen@reduc.edu.cu (I. d. l. M. L. d. Varona); julio.madera@reduc.edu.cu (J. Madera); analopezcorreoso@gmail.com (A. O. L. Correoso); laqdominguez@gmail.com (L. Q. Domínguez); ansel@cicese.edu.mx (A. Rodríguez-González); kristell.aguilar@plenitas.com (K. A. Alarcón); carlosdecastrolozano@gmail.com (C. d. C. Lozano); miguel@plenitas.com (J. M. R. Uceda)
0000-0002-1950-567X (Y. Martínez-López); 0000-0002-1886-7644 (I. d. l. M. L. d. Varona); 0000-0001-5551-690X (J. Madera); 0000-0002-3527-0516 (L. Q. Domínguez); 0000-0001-9971-0237 (A. Rodríguez-González); 0000-0001-8913-8388 (K. A. Alarcón); 0000-0001-6603-843X (C. d. C. Lozano); 0000-0002-5027-7521 (J. M. R. Uceda)

often include extensive training data that closely resembles the test sets, which may constrain the ability to genuinely evaluate the reasoning capabilities of language models on out-of-distribution inputs and to generalize across languages and proficiency regimes. The PROFE 2026 shared task, organized within the framework of IberLEF 2026 [4, 5, 6], addresses this gap by proposing to evaluate automatic systems under conditions comparable to those used for assessing human language learners. The task is grounded on real examinations produced by the Instituto Cervantes, the official institution responsible for the assessment and certification of Spanish as a foreign language, which evaluate proficiency at different reference levels of the Common European Framework of Reference for Languages (CEFR) [7]. Participating systems receive exercise items together with their original instructions, and must produce the correct answer in the same format expected from a human candidate. Unlike most traditional NLP benchmarks, PROFE 2026 deliberately does not provide task-specific training data: only a small number of complete exams are released as illustrative samples of the dataset, and systems are expected to rely on transfer learning approaches or on the zero-/few-shot capabilities of large generative language models (LLMs) to adapt to the task. This design choice both reflects the realistic scenario faced by an LLM-based pedagogical assistant deployed in production and prevents the common confound of test-set leakage through closely matched training data. The competition is structured into three complementary subtasks, each presenting distinct linguistic and reasoning challenges: Multiple Choice, in which the system must select the correct option among a closed set of candidates; Matching, in which the system must align elements from two parallel lists according to a semantic criterion; and Fill-in-the-Gap, in which the system must produce the lexical item that fits a given textual context. All three subtasks are evaluated through standard accuracy metrics computed at the item level. For teams that participate in the three subtasks of a complete exam, an additional exam-level score is computed based on the proportion of exams passed, where passing is defined as reaching at least 60% accuracy across all items of the exam, mirroring the threshold applied to human candidates, which enables direct and pedagogically meaningful comparisons between machine and human performance. In this paper, we describe our participation in PROFE 2026 with a methodological approach that combines prompt-based use of open-weight instruction-tuned LLMs with light-weight subtask-specific adaptation, designed to operate effectively under the data-scarce conditions imposed by the task.

2. Methodology

The PROFE 2026 shared task is formulated as a zero-shot reading comprehension problem over real examinations developed by the Instituto Cervantes for the certification of Spanish as a foreign language at different CEFR levels[8, 4]. The methodological proposal of our participation is built on three design principles that respond directly to the constraints of the task: (i) modular separation between the linguistic core of the pipeline (the LLM and embedding backends) and the sub-task-specific reasoning components, so that the same code base supports the three sub-tasks without architectural duplication; (ii) hybrid reasoning, in which a generative LLM is used as the primary solver when it returns a sufficiently confident answer and a sentence-embedding-based solver is used as a deterministic and always-available fallback; and (iii) strict compliance with the official submission format, so that the predictions of the three sub-tasks are directly evaluable by the organizers without any manual post-processing. The pipeline is implemented in Python 3 on top of PyTorch, HuggingFace sentence-transformers, scikit-learn, scipy and the native Ollama REST API, and runs on CUDA-enabled GPU environments when available or, alternatively, on CPU for the embedding components.

2.1. Subtasks

PROFE 2026 is structured into three complementary subtasks, one per exercise type, and teams can participate in any combination of them. Each subtask groups several exercises of the same type, drawn from real examinations developed by the Instituto Cervantes to certify Spanish as a foreign language. The three subtasks are defined as follows:

- **Multiple Choice (MC)**. Each exercise contains a textual passage and a set of multiple-choice questions about it, in which only one of the candidate options is correct. Given a question, the system must select the correct option among the candidates. Internally, each question is represented as a Question record carrying the question identifier, the question text, an optional image path, the list of options, the parent exercise and exam identifiers, the CEFR level and the surrounding contextual passage.
- **Matching**. Each exercise contains two sets of texts. The system must find, for every element of the first set, the element of the second set that best matches it. There is only one possible match per item, but the first set may contain extra elements that do not need to be matched, which means that the bijectivity assumption holds locally per matched pair but not globally over the whole set. Each exercise is represented as a MatchingExercise record carrying the exercise and exam identifiers, the CEFR level, the natural-language instructions and the two sets of items.
- **Fill-in-the-Gap (FG)**. Each exercise contains a text with several gaps, where the gaps correspond to textual fragments that have been removed and are presented disorderly as candidate options. The system must determine the correct position for each fragment. As in Matching, there is only one correct text per gap, but there may be more candidate fragments than gaps, so the assignment is partial: every gap is filled with exactly one fragment, but some fragments may remain unused. Each exercise is represented as a FillGapExercise record carrying the exercise and exam identifiers, the CEFR level, the gapped text (with gaps marked by <s id=...> tags), the list of gap identifiers and the list of candidate fillings.

The diversity of these three subtasks opens up research on how to approach them with different methodological strategies, including the design of subtask-specific prompts when generative models are used as the reasoning core.

Multimodal challenge. As the main novelty of the 2026 edition, some exercises contain images. The role of these images varies across instances and constitutes a genuine multimodal challenge:

- In certain exercises, the candidate answers themselves are images rather than text excerpts. The system must select the correct visual option among a closed set of images.
- In other exercises, images provide essential visual information that complements the textual prompt and is required to answer correctly (e.g., a question about a graph, a sign or a scene depicted in the image).
- Conversely, in a third class of exercises, the images are decorative and do not carry essential information for the answer; treating them as informative would mislead the system.

This heterogeneity introduces a fundamental new challenge with respect to the previous edition: participating systems must adopt a multimodal approach capable of discerning when to integrate visual cues and when to disregard them. The official PROFE 2026 test set distributed by the organizers contains 562 supporting images across the three subtasks, illustrating the practical scale of the multimodal component. In our current pipeline, this requirement is partially addressed through a text-first strategy with placeholder fallback: when an option or an item carries only an image path and no textual content, the corresponding text field is replaced by the placeholder and the embedding-based solver selects according to the textual context surrounding the visual option.

2.2. Evaluation Protocol

System outputs are evaluated by the organizers using accuracy as the main evaluation measure, defined as the proportion of correctly answered items. The choice of accuracy is methodologically motivated: in all three subtasks there is exactly one correct option among the candidates, and accuracy is also the measure applied to human candidates taking the same examinations. This dual applicability is the key methodological asset of PROFE 2026, because it enables a direct and pedagogically meaningful comparison between automatic systems and human learners under the same conditions. The evaluation is reported from two complementary perspectives:

- Question-level evaluation. Correct answers are counted individually across all items of a subtask, without grouping them by exercise or by exam. This perspective is applied to every team and every subtask separately, and it constitutes the primary ranking signal of the official leaderboard.
- Exam-level evaluation. Scores are computed at the granularity of complete exams. Each exam contains several exercises of different types (Multiple Choice, Matching and Fill-in-the-Gap), and an exam is considered passed if the accuracy obtained by the system across all items of that exam is above 0.6. The exam-level global score is then defined as the proportion of exams passed by the system. The 0.6 threshold deliberately mirrors the passing criterion applied to human candidates by the Instituto Cervantes, which is what makes machine-to-human comparison interpretable. This perspective is only applicable to teams that participate in the three subtasks simultaneously, since incomplete participation produces partial exams that cannot be evaluated under the same passing rule used for humans.

The specific accuracy formulations per subtask are:

- Multiple Choice: accuracy is the proportion of questions for which the predicted option identifier exactly matches the reference.
- Matching: accuracy is the proportion of correct text-to-text pairings in the predicted mapping, evaluated pair by pair against the reference mapping.
- Fill-in-the-Gap: accuracy is the proportion of gaps that are correctly filled, evaluated gap by gap against the reference assignment.

Internally, we use the same metrics on the development examples released by the organizers (the nine JSON files in the input/ folder, which do include reference answers) to compare backend configurations and prompt variants prior to selecting the final runs submitted to the task.

2.3. Dataset

The evaluation in PROFE 2026 is conducted on a partition of the IC-UNED-RC-ES corpus [4], a collection of reading comprehension exercises drawn from the official examinations developed by the Instituto Cervantes (IC) for the certification of Spanish as a foreign language at different reference levels of the Common European Framework of Reference for Languages (CEFR). The corpus is jointly curated by the Instituto Cervantes and the Universidad Nacional de Educación a Distancia (UNED) and is distributed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) licence, which authorizes its use for non-commercial research purposes while preserving the attribution and share-alike obligations on derivative works. The use of authentic Instituto Cervantes examinations is a defining methodological feature of PROFE 2026 and one of the principal sources of difficulty of the task. Unlike most NLP benchmarks, in which a system is evaluated against synthetic or crowd-sourced material constructed specifically for machine evaluation, here automatic systems are confronted with the same items that real human candidates face in official certification examinations. This design choice has three direct consequences for the methodology of any participating system. First, the items have been calibrated for human difficulty across CEFR levels, which means that they probe a broad spectrum of linguistic and pragmatic competences, vocabulary, inference, register, discourse cohesion, cultural knowledge— rather than narrow surface patterns. Second, the linguistic style is the careful, edited Spanish of an official assessment, with idiomatic and culturally specific items that are scarce in standard pretraining corpora dominated by web text. Third, and most importantly, the comparability between machine and human scores is direct and pedagogically meaningful. The dataset is distributed in two complementary folders, each one playing a distinct role in the development cycle of a participating system:

- A development folder (input/) containing nine JSON files that include the reference answers for the three subtasks. This folder is released as a worked illustration of the official data format and is intended for understanding the structure of the items, validating the pipeline end to end,

and performing local accuracy estimates on a small held-out sample before producing the final submission. In our pipeline, this folder is used exclusively as a sanity-check partition for backend selection and prompt-variant comparison.

- An official test folder (`PROFE26_test_set/`) containing three JSON files (one per subtask: `multiple_choice_dataset.json`, `matching_dataset.json`, `fill_the_gap_dataset.json`) plus 562 supporting images that may appear as candidate options or as contextual material in the exercises. The reference answers are deliberately withheld in this partition; all evaluation on the test set is performed by the organizers after submission. The presence of the images materializes the multimodal challenge of the 2026 edition described in Section 2.1.

Our data loader processes each subtask file by iterating over the exams field, extracting the exercises and items, and constructing the corresponding internal Question, MatchingExercise and FillGapExercise records Image paths are resolved at load time relative to the dataset root, and texts are preserved verbatim without lowercasing, accent stripping or punctuation normalization, since reading comprehension at CEFR levels depends critically on surface features such as accent marks, punctuation conventions and morphological inflection that would be lost under aggressive normalization. Records that cannot be parsed (e.g., due to missing required fields) are silently skipped and logged for diagnostic purposes, but in practice the official PROFE 2026 test set parses cleanly under this loader.

2.4. Sentence Embeddings

For all three subtasks, our pipeline relies on a multilingual sentence-embedding backbone that maps texts into a continuous space where geometric proximity reflects semantic similarity. Embeddings are dense, low-dimensional vector representations that convert symbolic data intrinsically sparse and combinatorial into numerical objects on which linear algebra, cosine-based similarity metrics and gradient-free assignment algorithms can operate efficiently [9, 10]. This shift from symbolic to distributed representations is one of the foundational ideas of modern Natural Language Processing and lies behind most current systems in information retrieval, recommendation, classification and clustering. As the primary encoder of our pipeline we adopted intfloat/multilingual-e5-large-instruct, an instruction-tuned multilingual encoder that produces dense 1024-dimensional embeddings with strong cross-lingual alignment and explicit support for Spanish. The encoder is loaded through the sentence-transformers library with `trust_remote_code=True`, runs on GPU when available, and encodes inputs in batches of 32. During development we additionally explored two alternative encoders that the literature reports as strong on multilingual retrieval and semantic similarity benchmarks:

- nomic-ai/nomic-embed-text-v1, a fully open-source long-context encoder (up to 8192 tokens) trained with contrastive learning on a large-scale web corpus, designed to provide competitive performance on retrieval and clustering benchmarks while remaining lightweight enough for local deployment.
- sentence-transformers/paraphrase-multilingual-mpnet-base-v2, a multilingual MPNet-based encoder widely used as a strong baseline for cross-lingual semantic similarity.

The final submission was produced with multilingual-e5-large-instruct, which exhibited the most stable behaviour on the development examples across the three subtasks. The choice of an instruction-tuned encoder is deliberate: PROFE 2026 examinations consistently frame items as natural-language instructions (e.g., "select the correct option", "match the descriptions"), and an encoder trained to interpret such instructions aligns more naturally with the inputs than a purely contrastive one.

2.5. Large Language Models

In addition to the embedding backbone, our pipeline integrates an open-weight instruction-tuned LLM as the primary solver for the Multiple Choice subtask. Large Language Models are deep neural

networks —overwhelmingly based on the Transformer architecture— trained on massive amounts of text under a self-supervised next-token prediction objective. Despite the apparent simplicity of this training signal, scaling these models to billions of parameters and trillions of tokens has produced systems that exhibit broad linguistic competence, multi-step reasoning, and the ability to follow natural-language instructions across tasks for which they were not explicitly trained. For PROFE 2026, where no task-specific training data is provided and the items are framed as instructions to a human learner, this instruction-following capability is precisely the property that the system needs to exploit. We evaluated three open-weight instruction-tuned LLMs released between 2024 and 2025, served locally through Ollama:

- Qwen2.5-7B, developed by Alibaba Cloud, an instruction-tuned multilingual model trained on a corpus of approximately 18 trillion tokens with explicit Spanish coverage.
- Llama-3.1-8B-Instruct, developed by Meta, a widely adopted multilingual baseline
- Gemm4-cloud, developed by Google DeepMind, an open-weight model with refined alignment procedures and competitive performance on multilingual reasoning benchmarks.

The integration with Ollama is encapsulated in a single OllamaSolver class that supports three deployment modes through the same interface: a local mode against an Ollama daemon running at http://localhost:11434 with no authentication, a cloud mode against https://ollama.com with Bearer authentication via an API key (read explicitly or from the `OLLAMA_API_KEY` environment variable), and an arbitrary remote mode against any user-supplied Ollama-compatible endpoint. The class performs a connectivity check at construction time against the /api/tags endpoint, and dispatches each prediction as a POST to /api/generate with stream=False, the configured model identifier, and decoding options set to temperature = 0.1. The deliberately low temperature is motivated by the deterministic nature of the task: examination items have a single correct answer, and lexically conservative, near-greedy decoding is systematically preferred over creative paraphrases. For the Multiple Choice subtask, the LLM is prompted with a structured Spanish template that explicitly declares the role of the model ("Eres un experto en comprensión lectora del español"), provides the contextual passage, the question and the enumerated options, and instructs the model to respond only with the letter of the correct option (A, B, C or D). The returned response is parsed character by character; the first character belonging to the valid option set is taken as the prediction, with a confidence score of 0.8 if it appears within the first five characters of the response (signalling a clean answer) and 0.6 otherwise. If no valid character is found, the system falls back to a default prediction with confidence 0.3.

2.6. Subtask-Specific Solvers and the Hungarian Ensemble

The reasoning core of the pipeline is implemented as three subtask-specific solvers that share the embedding backbone and, in the case of Multiple Choice, also share the LLM backend: Multiple Choice solver. A hybrid LLM-then-embedding solver. The LLM is invoked first through the structured prompt described in Section 2.5; if it returns a valid option with confidence above 0.5, that answer is committed. Otherwise, the system falls back to a deterministic embedding-based solver: the concatenation of the context and the question is encoded into a single query vector, each option text is encoded independently, and the option with the highest cosine similarity to the query is selected. The fallback similarity is rescaled from [1,1][-1, 1] [1,1] to [0,1][0, 1] [0,1] to produce a confidence score that is comparable to the LLM confidence and that can be used for downstream analysis. Matching solver. A pure embedding-based solver formulated as an optimal assignment problem. The two sets of items (set1, set2) are encoded independently into matrices of sentence embeddings, and a pairwise similarity matrix:

$$S \in \mathbb{R}^{|\text{set}_2| \times |\text{set}_1|} \tag{1}$$

, is computed via inner product. The matching is then obtained by solving

$$\min_{\pi} \; -\sum_{i} S_{i,\pi(i)} \tag{2}$$

through the Hungarian algorithm (`scipy.optimize.linear_sum_assignment`), which guarantees a global optimum in $O(n^3)$ time over the cost matrix -S. This formulation enforces a bijective mapping between the two sets, which is the structural constraint of the official Matching subtask and which a naïve argmax-per-row approach would systematically violate. "Fill-in-the-Gap solver" A pure embedding-based solver that exploits "local textual coherence". For each gap marked by ‘<s id=...>‘, a window of 100 characters is extracted before and after the marker, producing a *gap context* of the form ‘<before> [GAP] <after>‘. Each gap context is then encoded into a single vector, each candidate filling is encoded independently, and a coherence matrix

$$C \in \mathbb{R}^{|\text{gaps}| \times |\text{fillings}|} \tag{3}$$

, is computed via inner product. As in the Matching solver, the final assignment is obtained by applying the Hungarian algorithm to the negated coherence matrix, which enforces a "bijective gap-to-filling mapping" and prevents the system from selecting the same filling for two different gaps —a violation that, in our experience with low-resource generative LLMs prompted under high lexical pressure, is otherwise frequent. The three solvers can therefore be viewed as a lightweight ensemble of complementary reasoning operations: hybrid LLM-and-embedding similarity for the open-ended question regime (Multiple Choice), and embedding-based optimal-assignment for the two structurally constrained regimes (Matching and Fill-in-the-Gap). The choice of the Hungarian algorithm as the assignment backbone for the latter two subtasks is methodologically important: it converts what would otherwise be a sequence of locally greedy decisions into a single globally optimal one with respect to the chosen semantic similarity, and it removes from the LLM the burden of enforcing the bijectivity constraint, which is a known weakness of prompt-based LLMs in structured prediction.

2.7. System Architecture

The pipeline is organized as a sequential composition of six decoupled modules, which makes it possible to replace individual components without affecting the rest:

- Module 1 — Data ingestion. Reads the three official JSON files (Multiple Choice, Matching, Fill-in-the-Gap) from the configured `DATA_DIR`, resolves image paths, and constructs the typed in-memory records (Question, MatchingExercise, FillGapExercise) consumed by the rest of the pipeline.
- Module 2 — Embedding backbone. Loads the configured sentence-transformers model (multilingual-e5-large-instruct in the final submission) onto the available device and exposes a single encode(texts) method that returns L2-normalizable dense vectors in batches.
- Module 3 — LLM provider. Wraps the Ollama REST API behind the OllamaSolver class, with support for local, remote and cloud deployment modes (Section 2.5). Exposes a single `solve_multiple_choice`(context, question, options) method that returns an (`answer_id`, confidence) tuple.
- Module 4 — Subtask solvers. Implements the three solvers: `solve_multiple_choice` (hybrid LLM/embedding), `solve_matching` (Hungarian over similarity), `solve_fill_gap` (Hungarian over coherence). Each solver returns the data structure expected by the official submission format.
- Module 5 — Inference loop. Iterates over the three datasets through tqdm-instrumented loops, calls the appropriate solver for each instance, and accumulates the predictions into three dictionaries (`mc_results`, `matching_results`, `fill_gap_results`).
- Module 6 — Submission writer. Serializes the three prediction dictionaries to UTF-8 JSON files (`multiple_choice_preds.json`,`matching_preds.json`,`fill_the_gap_preds.json`) and packages them into a single submission.zip archive that complies with the official submission format.

The full configuration —data directory, output directory, embedding model identifier, Ollama deployment mode, model identifier, API key and inference device— is centralized in a single configuration block at the head of the entry point, making any experiment fully reproducible from its invocation environment.

3. Template parameters

Figure 1 Summarizes the end-to-end workflow of our PROFE 2026 pipeline, organized as six sequential stages that mirror the modular structure of the implementation. In the first stage, the data ingestion module reads the three official JSON files (`multiple_choice_dataset.json`, `matching_dataset.json`, `fill_the_gap_dataset.json`) from the configured `DATA_DIR`, resolves image paths and constructs the typed in-memory records (Question, MatchingExercise, FillGapExercise) consumed by the rest of the pipeline.

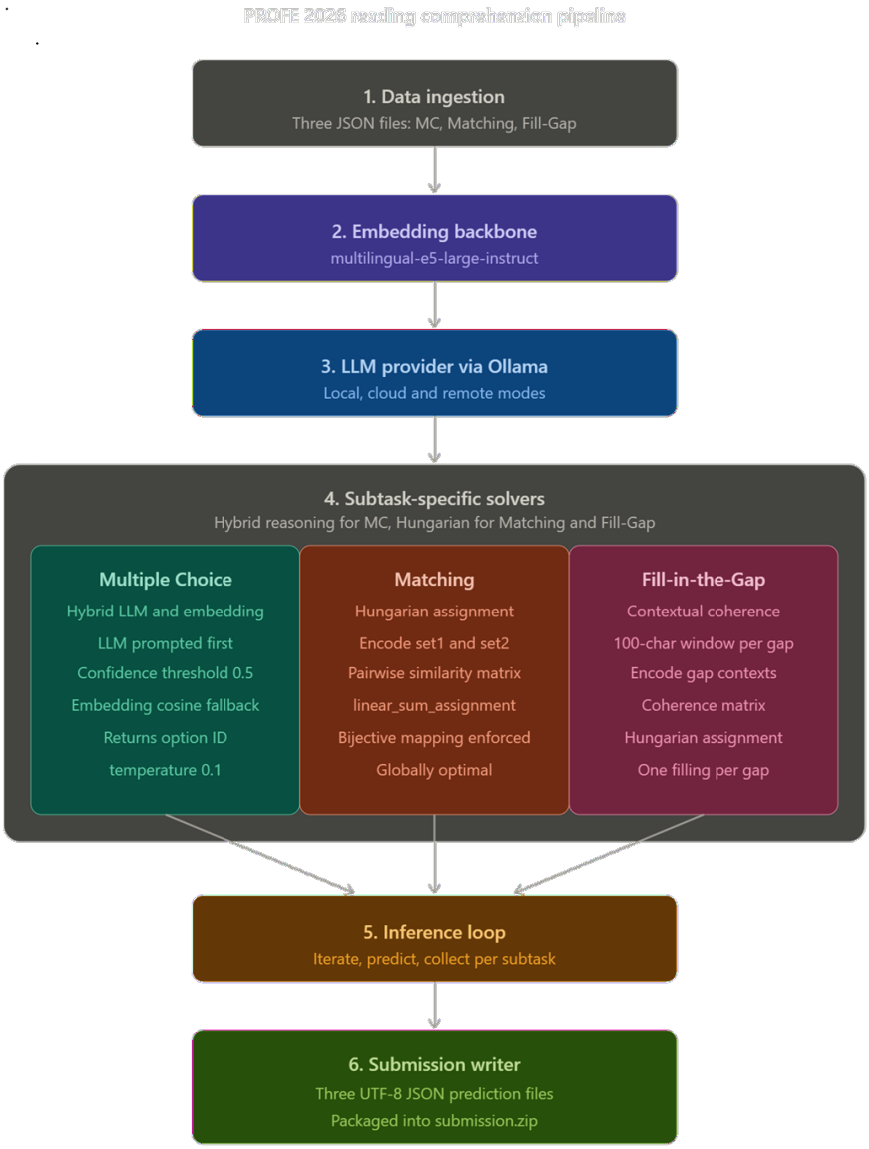

Figure 1: Summarizes the end-to-end workflow of the pipeline

The second stage loads the embedding backbone (intfloat/multilingual-e5-large-instruct) through the sentence-transformers library and exposes a single encode(texts) method that produces dense 1024-dimensional vectors in batches. The third stage initializes the LLM provider, an OllamaSolver instance that supports local, cloud and arbitrary remote deployment modes behind a single class and performs a connectivity check against the /api/tags endpoint at construction time. The fourth stage —the reasoning core of the pipeline— branches into three subtask-specific solvers that share the embedding backbone and, in the case of Multiple Choice, also share the LLM backend: a hybrid

LLM-then-embedding solver for Multiple Choice, in which the LLM is prompted first with temperature = 0.1 and a structured Spanish template, and the embedding-based cosine-similarity solver is used as a fallback when the LLM does not return a confident answer; a Hungarian-based optimal-assignment solver for Matching, which guarantees a bijective mapping between the two sets of items through `scipy.optimize.linear_sum_assignment`; and a Hungarian-based coherence solver for Fill-in-the-Gap, which extracts a 100-character window around each gap, encodes the resulting contexts into a coherence matrix against the candidate fillings, and applies the same optimal-assignment formulation to obtain a globally optimal gap-to-filling mapping. Regardless of which subtask is processed, the pipeline converges in the fifth stage, where the inference loop iterates over the corresponding dataset through tqdm-instrumented loops, invokes the appropriate solver for each instance and accumulates the predictions into per-subtask dictionaries with explicit handling of generation failures so that a single failed instance never aborts the full test pass. Finally, the sixth stage serializes the three prediction dictionaries to UTF-8 JSON files (`multiple_choice_preds.json`, `matching_preds.json`, `fill_the_gap_preds.json`) and packages them into a single submission.zip archive that complies with the official PROFE 2026 submission format, ensuring that the output is directly evaluable by the organizers without any manual post-processing.

4. Results

Table 1 reports the internal comparison of five configurations combining a generative backbone with a sentence-embedding retriever, evaluated across three question formats: Multiple choice, Matching, and Filling, and an aggregated Exam level score. The configurations were selected to isolate two factors: (i) the effect of the generative model, by contrasting gemma4:31b-cloud against smaller open-weights models (qwen2.5:7b, llama-3.1-8B-Instruct), and (ii) the effect of the embedding model used for context retrieval, by pairing the strongest generator with three alternatives (multilingual-e5-large-instruct, multilingual-e5-large, and paraphrase-multilingual-mpnet-base-v2). Overall ranking. The configuration gemma4:31b-cloud + multilingual-e5-large-instruct achieves the highest score on every question format and on the aggregated Exam level metric (93.44, 61.51, 38.60, and 81.01, respectively). It outperforms the second-best configuration by a substantial margin at the exam level (+8.38 points over gemma4:31b-cloud + paraphrase-multilingual-mpnet-base-v2), suggesting that the combination of a stronger generator with an instruction-tuned multilingual retriever provides complementary gains rather than redundant ones. Effect of the generative model. Holding the embedding model fixed at nomic-embed-text-v1, qwen2.5:7b and llama-3.1-8B-Instruct perform almost identically (63.13 vs. 61.45 at the exam level, with a gap of only 1.68 points), indicating that, at this parameter scale, model family has a limited impact on downstream performance for this task. In contrast, replacing the 7–9B-parameter backbone with gemma4:31b-cloud, while keeping a comparable embedding model yields a much larger improvement (e.g., from 63.13 to 81.01 at the exam level when paired with multilingual-e5-large-instruct), which confirms that scaling the generator is the single most influential factor in the pipeline. Effect of the embedding model. Fixing the generator at gemma4:31b-cloud, the choice of retriever produces non-trivial differences. multilingual-e5-large-instruct clearly dominates (81.01), followed by paraphrase-multilingual-mpnet-base-v2 (72.63) and the non-instruct variant multilingual-e5-large (70.39). The 10.62-point gap between the instruct and non-instruct E5 variants is particularly noteworthy, as both share the same architecture and pretraining corpus and differ only in the instruction-tuning stage. This supports the hypothesis that instruction-tuned retrievers are better aligned with the semantics of exam-style queries, where the prompt explicitly describes the type of information being sought. Performance by question format. All configurations follow the same difficulty ordering: Multiple choice > Matching > Filling. Filling items are markedly harder for every system (best score 38.60, worst 33.33), which is consistent with the open-ended nature of the format: unlike Multiple choice or Matching, fill-in-the-blank items cannot be solved by ranking a closed set of candidates and therefore penalise both retrieval imprecision and generative hallucination. The fact that even the best configuration remains below 40 points in this column suggests that Filling is the principal bottleneck of the pipeline and the most promising direction

for future improvement.

Table 1
Internal Comparisons

Method	Multiple choice	Matching	Filling	Exam level
gemma4:31b-cloud+multilingual-e5-large-instruct	93.44	61.51	38.6	81.01
qwen2.5:7b+nomic-embed-text-v1	73.66	58.1	35.96	63.13
llama-3.1-8B-Instruct+nomic-embed-text-v1	73.61	57.06	33.33	61.45
gemma4:31b-cloud+multilingual-e5-large	80.82	56.75	35.96	70.39
gemma4:31b-cloud+paraphrase-multilingual-mpnet-base-v2	84.92	56.75	35.96	72.63

The results indicate that, within the explored design space, the dominant lever is the capacity of the generative model, with the embedding model acting as a secondary but non-negligible— amplifier when it is instruction-tuned. Based on this analysis, gemma4:31b-cloud + multilingual-e5-large-instruct was selected as the final configuration submitted to the shared task. Table 2 summarizes the official evaluation results released by the shared-task organizers. Three teams submitted runs to the task: UCO-UC-CICESE-Plenitas (our system), NIL_UCM, and David, although not all of them produced predictions for every question format, which results in a sparse comparison matrix. Overall performance. Our submission, UCO-UC-CICESE-Plenitas, was the only system to provide predictions for the three question formats simultaneously and is therefore the only one with an aggregated Exam level score (81.01). This already constitutes a relevant outcome of the task, since the Exam level metric is designed to reward systems capable of handling the full diversity of item types encountered in a real proficiency exam, rather than specializing in a single format. This is the only format with predictions from more than one team, and consequently the only one where a direct ranking can be established. Our system obtains the best result (93.44), followed by the two NIL_UCM runs (90.55 and 88.85). The margin over the closest competitor is 2.89 points, and 4.59 points over the second NIL_UCM run. Although the gap is moderate, it is consistent with the trend observed in our internal ablations (Table **??**), where the combination of gemma4:31b-cloud with an instruction-tuned multilingual retriever proved particularly effective for closed-set classification items. Comparison on Matching. On this format, the second NIL_UCM run obtains the highest score (86.51), substantially above the submission by David (66.67) and our own system (61.51). This is the only format on which our pipeline is clearly outperformed, and the gap (25.00 points with respect to the best run) deserves a dedicated discussion. A plausible explanation is that Matching items require establishing a global alignment between two sets of elements, a task that does not map naturally onto the per-item retrieval-then-generation strategy adopted by our pipeline. NIL_UCM's ability to specialize on this format at the cost of not submitting Filling or Exam level runs suggests that a dedicated matching-aware decoding strategy could yield significant gains, and we leave this as a clear direction for future work. Our system is the only participant to submit predictions for the fill in the blank format, obtaining a score of 38.60. While no direct comparison is possible, the absolute value is consistent with the difficulty of the format and confirms that open-ended generation remains the hardest sub-task of the benchmark for all participants, to the point that two of the three teams opted not to submit runs for it. A noticeable feature of the official results is the sparsity of the comparison matrix: out of twelve possible (team, format) cells, only seven are populated. This pattern suggests that several participants adopted a format-specialized strategy, optimizing their system for one or two formats rather than building a unified pipeline. In contrast, our approach was deliberately designed as a single end-to-end pipeline applicable to all question types, which, although it does not lead to the best score on Matching, results in the only system with a complete and competitive coverage of the exam, and consequently the only one eligible for the Exam level ranking.

The official evaluation positions UCO-UC-CICESE-Plenitas as the best system on Multiple choice and the only system with a full Exam level submission (81.01), while identifying Matching as the format with the largest margin for improvement. These results validate the design choices reported in the previous section and, at the same time, highlight a concrete avenue for future research on dedicated

Table 2
Official Evaluation Results

Team/Participant	Multiple choice	Matching	Filling	Exam level
UCO-UC-CICESE-Plenitas	**93.44**	61.51	**38.6**	**81.01**
NIL_UCM	90.55			
NIL_UCM	88.85	86.51		
David		66.67		

matching-oriented decoding.

5. Conclusions

This paper has described the UCO-UC-CICESE-Plenitas system submitted to PROFE 2026, a shared task that evaluates automatic systems against the same Instituto Cervantes examinations used to certify human learners of Spanish as a foreign language. Our methodological proposal was built around three design principles —modular separation of backends and reasoning, hybrid LLM-and-embedding reasoning, and strict compliance with the official submission format— and was deliberately designed as a single end-to-end pipeline applicable to the three subtasks rather than as a collection of format-specialized systems. The internal evaluation on the development partition allowed us to isolate the contribution of each component of the pipeline. The combination of gemma4:31b-cloud with multilingual-e5-large-instruct consistently dominated every other configuration across the three subtasks and at the exam level, with an improvement of more than 17 absolute points over the strongest 7-9B-parameter alternative. Two findings deserve particular attention. First, scaling the generator is the single most influential factor in the pipeline: replacing a mid-size open-weights backbone with gemma4:31b-cloud yielded substantially larger gains than any change on the retriever side. Second, instruction-tuning of the retriever matters: the 10.62-point gap between the instruct and non-instruct variants of E5, which share architecture and pretraining corpus and differ only in the instruction-tuning stage, supports the hypothesis that instruction-tuned encoders are better aligned with the semantics of exam-style queries. The official evaluation confirmed the practical value of the unified design. Our system was the only one to provide predictions for the three sub-tasks simultaneously and, consequently, the only one eligible for the Exam-level ranking (81.01). It also obtained the best result on Multiple Choice (93.44), 2.89 points above the closest competitor. However, in Matching, our system was clearly outperformed by a specialized submission (61.51 vs. 86.51); we attribute this gap to the fact that Matching requires a global alignment between two sets of elements, a structural problem on which a retrieval-then-generation strategy is at a disadvantage. On Fill-in-the-Gap, the absolute value of 38.60 confirms that open-ended generation remains the hardest regime of the benchmark, to the point that two of the three participating teams opted not to submit runs for this format. The principal known limitation of our submission is the absence of a true multi-modal component. The 2026 edition of PROFE introduced 562 supporting images across the test set, in which the role of the visual signal ranges from being the answer itself, to providing essential complementary information, to being purely decorative. Our pipeline addresses this challenge only partially, through a text-first strategy with a placeholder fallback that preserves the structural integrity of the bijective assignments in Matching and Fill-in-the-Gap but does not exploit the visual information itself. The Matching gap observed on the official evaluation is, to a significant extent, a reflection of this limitation. These observations outline three concrete directions for future work. First, the integration of vision-language models (e.g., CLIP or a multi-modal Ollama-served LLM) to encode visual options into the same embedding space used for textual ones, which we expect to recover a substantial fraction of the gap currently observed on items involving images. Second, the exploration of matching-aware decoding strategies,for example, jointly fine-tuning the retriever on assignment-style objectives, or replacing the linear assignment formulation by a constrained-decoding LLM step to close the gap on the Matching sub-task. Third,

the strengthening of the Fill-in-the-Gap solver, which we identify as the principal bottleneck of the pipeline, possibly through cloze-style fine-tuning of the embedding encoder or through a generative re-ranker over the candidate fillings. Overall, the results obtained on PROFE 2026 validate the modular, hybrid design adopted by our team and support its use as a robust baseline for future editions of the task, while clearly outlining the multi-modal and structural extensions that would be required to close the remaining gaps.

Declaration on Generative AI

During the preparation of this work, the authors used Claude(Opus 4.7) and Grammarly in order to: Figure, Grammar and spelling check. After using these services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] A. Zubiaga, Natural language processing in the era of large language models, Frontiers in Artificial Intelligence 6 (2024) 1350306.

[2] B. MacCartney, Natural Language Inference, Ph.D. thesis, Stanford University, Stanford, CA, 2009.

[3] G. Lai, Q. Xie, H. Liu, Y. Yang, E. Hovy, RACE: Large-scale reading comprehension dataset from examinations, 2017. URL: https://arxiv.org/abs/1704.04683. arXiv:1704.04683.

[4] Á. 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).

[5] 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 Workshop Proceedings, CEUR-WS.org, 2026.

[6] Y. Martínez-López, M. G. Saborit, Y. Jauriga, M. L. D. Varona, J. Madera, A. Rodríguez-González, J. C. A. Fernández, UC-UCO-CICESE_UT3-Plenitas team—exploring in the PROFE 2025: Language proficiency evaluation, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[7] Á. Rodrigo, S. Moreno-Álvarez, A. P. García-Plaza, A. Peñas, R. Agerri, J. Fruns-Jiménez, I. Soria-Pastor, Overview of PROFE at IberLEF 2025: Language proficiency evaluation, Procesamiento del Lenguaje Natural 75 (2025).

[8] A. Peñas, Á. Rodrigo, J. Fruns-Jiménez, I. Soria-Pastor, S. Moreno-Álvarez, A. Pérez, J. Reyes-Montesinos, A Spanish language proficiency dataset for AI evaluation, Information 17 (2026) 159. URL: https://doi.org/10.3390/info17020159. doi:10.3390/info17020159.

[9] Q. Liu, M. J. Kusner, P. Blunsom, A survey on contextual embeddings, arXiv preprint arXiv:2003.07278 (2020).

[10] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, W. Wang, Language-agnostic BERT sentence embedding, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2022, pp. 878–891.

A. Online Resources

The results of the PROFE 2026 are available via:

- PROFE 2026,

- PROFE 2026 Results,
- PROFE 2026 Code