Few-Shot LLM-as-Judge with Language-Aware Translation for Multilingual Answer Scoring

ELOQUENT Sensemaking at CLEF 2026

Dihan Dai[1], Jiajian Lu[1] and Jing Zhang[1]

[1]*Independent Researcher*

Abstract

We (team **ASTeam**) describe our approach to the CLEF 2026 ELOQUENT Sensemaking shared task, which requires automated scoring of student answers across 13 languages using both rubric-based (3-level) and simple (5-level) rating scales. Our system uses Qwen3-32B as an LLM-based evaluator with three key enhancements: (1) domain-matched few-shot examples retrieved from the training set, (2) language-aware evaluation with explicit inline translation of non-English answers before scoring, and (3) deterministic (greedy, temperature=0) decoding for run-to-run consistency from a single inference pass. The same method is applied to both tracks with minimal modification. On the development set, our approach achieves QWK scores of 0.821 (rubric track) and 0.607 (simple track), substantially outperforming the Llama-4-Scout-17B baseline of 0.760 and 0.360, respectively. Error analysis reveals that extreme rating errors (0↔2 confusions) are reduced to under 1% of items through the combination of few-shot calibration and translate-then-evaluate prompting.

Keywords

automated essay scoring, LLM-as-judge, few-shot learning, multilingual NLP, CLEF 2026

1. Introduction

The ELOQUENT Sensemaking shared task at CLEF 2026 challenges participants to develop automated systems for scoring student answers across multiple languages and domains. The task features two tracks: (1) Rubric-Based Rating, where answers are scored on a 0–2 scale using explicit rubric descriptions, and (2) Simple Rating, where answers are scored on a 0–4 scale without rubric guidance. Both tracks span 13 languages and diverse educational domains, making this a challenging multilingual evaluation problem.

Recent work has demonstrated the effectiveness of large language models (LLMs) as evaluators [1, 2], with LLM-based judges achieving strong correlation with human ratings across various assessment tasks. However, multilingual scoring introduces additional challenges: the model must understand answers in the source language while comparing them against English-language rubrics, and scoring standards must remain calibrated across linguistic and cultural boundaries.

Prior work on the CLEF 2025 Sensemaking task [3] examined LLMs in the evaluator role and highlighted the difficulty of reliably distinguishing topically relevant but substantively incorrect answers from correct ones. This failure mode—conflating semantic similarity with factual correctness—manifests as extreme errors (e.g., scoring "No Credit" answers as "Full Credit"), which are heavily penalized by Quadratic Weighted Kappa (QWK).

Our approach addresses these challenges through three complementary techniques:

1. **Domain-matched few-shot examples** from the training set anchor the model's understanding of the scoring scale for each specific question type, reducing extreme label confusions.
2. **Language-aware evaluation with inline translation** forces the model to explicitly process the semantic content of non-English answers before comparing against the English rubric, preventing surface-level pattern matching.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

dihandai2022@gmail.com (D. Dai); lujiajian1229@gmail.com (J. Lu); jingz3017@gmail.com (J. Zhang)

3. **Deterministic decoding** (greedy, temperature=0) yields consistent run-to-run outputs from a single inference pass, trading the variance estimate one would obtain from repeated sampling for a substantial reduction in inference cost.

We note that Qwen3-32B is a reasoning model operated in thinking mode, and that the task guidelines recommend running reasoning models multiple times so that the variability of their outputs can be characterized. Our single-pass deterministic setup is therefore a deliberate cost–variance trade-off rather than a complete solution: it gives reproducible results for a fixed endpoint and prompt, but it does not quantify the run-to-run uncertainty that multiple sampled runs would expose. We discuss this limitation further in Section 7.4.

We apply the same method architecture to both the rubric-based and simple rating tracks, demonstrating that a unified approach can handle different scoring granularities without task-specific engineering. Our code and prompts are publicly available.[1]

2. Related Work

2.1. LLM-as-Judge

The use of large language models (LLMs) as automated evaluators has garnered considerable scholarly attention in recent years. Zheng et al. [1] demonstrated that GPT-4 achieves agreement rates exceeding 80% with human annotators on open-ended generation tasks, thereby establishing the empirical viability of LLM-based evaluation frameworks. Liu et al. [2] further showed that incorporating chain-of-thought reasoning [4] prior to score assignment yields improved alignment with human judgments, providing a theoretical motivation for our approach of eliciting explicit reasoning traces in conjunction with numerical ratings.

2.2. Few-Shot In-Context Learning

Brown et al. [5] established that large language models are capable of generalizing from a limited number of contextually provided examples. In the context of scoring tasks, few-shot exemplars function as calibration anchors, enabling the model to internalize the intended granularity and boundaries of the rating scale. The present work extends this paradigm by aligning exemplars with the domain of the target item, thereby providing task-relevant calibration that more accurately reflects the evaluative context.

2.3. Multilingual Evaluation

Evaluation across multiple languages presents a distinctive set of challenges, particularly given that scoring rubrics are conventionally authored in English while student responses may be produced in any of 13 target languages. The ELOQUENT Sensemaking task [3] situates this problem in an educational-assessment setting, where evaluators must compare cross-lingual responses against English-language criteria. A natural strategy is to translate responses into the rubric language before scoring, so that the comparison between a student response and its corresponding rubric is conducted within a shared linguistic space. Rather than treating translation as a discrete preprocessing step, the proposed framework integrates it directly into the evaluation prompt, streamlining the pipeline and reducing the potential for information loss introduced by separate translation stages.

3. Task Description

The CLEF 2026 ELOQUENT Sensemaking shared task furnishes a comprehensive dataset of student-generated responses encompassing 13 typologically diverse languages — namely Czech, Danish, English,

[1] https://github.com/DihanDai/Sensemaking2026ASTeam-github

Finnish, German, Greek, Hungarian, Irish, Portuguese, Romanian, Serbian, Swedish, and Ukrainian — drawn from a broad range of educational domains.

3.1. Rubric-Based Rating Track

Each item consists of:

- A **question** describing the assessment task
- A student **answer** (in one of 13 languages)
- A set of **rubric descriptions** defining the criteria for each score level
- The target **label**: 0 (No Credit), 1 (Partial Credit), or 2 (Full Credit)

3.2. Simple Rating Track

Each item in the dataset comprises a question, a corresponding student response, and a target label drawn from an ordinal scale ranging from 0 to 4. Notably, no explicit rubric descriptions are provided, necessitating that the model infer appropriate scoring criteria solely from the contextual information embedded within the question itself.

3.3. Evaluation Metric

The primary evaluation metric employed in this work is Quadratic Weighted Kappa (QWK) [6], a measure of inter-rater agreement that applies quadratic penalization to disagreements of greater magnitude. QWK is particularly well-suited to ordinal rating scales, as it imposes substantially heavier penalties on extreme misclassifications (e.g., 0↔2 or 0↔4) relative to disagreements between adjacent categories (e.g., 0↔1).

The quadratic weight matrix is formally defined as:

$$w_{ij} = \frac{(i-j)^2}{(N-1)^2} \tag{1}$$

where N denotes the total number of rating categories. In the context of the rubric track ($N = 3$), a misclassification spanning two categories (e.g., 0↔2) incurs a penalty four times greater than that associated with a single-category disagreement (e.g., 0↔1).

3.4. Dataset Statistics

The development set comprises 4,146 annotated items, each accompanied by both rubric-based and simple track labels. The test set encompasses approximately 93,500 items distributed across both evaluation tracks. The training set, which provides labeled exemplars, is leveraged in this work to support few-shot example retrieval during the scoring process.

4. Method

4.1. Model Selection

We selected Qwen3-32B [7], a 32-billion parameter dense language model accessed via Amazon Bedrock [8], as the backbone evaluator for this work. The selection process involved a systematic comparative evaluation of candidate models on a stratified 50-item sample drawn from the development set, the results of which are summarized in Table 1.

Qwen3-32B was ultimately selected on the basis of its superior QWK score, a zero-percent extraction failure rate, and competitive inference latency relative to the other evaluated candidates. It is worth noting that despite achieving comparatively high raw accuracy, DeepSeek R1 yields a substantially degraded QWK score, attributable to a disproportionate frequency of extreme scoring errors; specifically,

Table 1
Model comparison on 50-item development sample (Rubric track)

Model	QWK	Failure Rate	Latency (s/item)
Qwen3-32B	**0.804**	0%	1.2
GPT-OSS-120B	0.741	2%	2.1
DeepSeek R1	0.573	10%	3.4
Claude Haiku 4.5	0.253	0%	3.7

6 out of 50 items exhibited an absolute deviation of $|pred - ref| = 2$. This outcome underscores the importance of adopting QWK as the primary selection criterion during model evaluation, as raw accuracy alone may obscure the practical cost of large ordinal misclassifications.

4.1.1. Architectures and Pretraining of the Candidate Models

The candidate models differ substantially in architecture, scale, and pretraining procedure, which is relevant context for interpreting the comparison in Table 1. Table 2 summarizes their salient properties.

Table 2
Architectural and pretraining characteristics of the candidate models. "Active params" denotes parameters activated per token (equal to total parameters for dense models). Pretraining-token counts are as reported by the model providers.

Model	Type	Total / Active	Context	Pretrain tokens	Reasoning
Qwen3-32B [7]	Dense	32.8B / 32.8B	32K (128K YaRN)	36T	Yes (toggle)
GPT-OSS-120B [9]	MoE	117B / 5.1B	128K	n/r	Yes
DeepSeek-R1 [10]	MoE	671B / 37B	128K	n/r (V3-base)	Yes
Llama-4-Scout [11]	MoE	109B / 17B	10M	∼40T	No
Claude Haiku 4.5 [12]	Proprietary	n/d	200K	n/d	Yes (toggle)

The models span three architectural families. **Qwen3-32B** is a dense, 64-layer decoder-only transformer that uses grouped-query attention (GQA, 64 query heads and 8 key/value heads) with QK-Norm for training stability, and a 151,669-token byte-level BPE vocabulary [7]. It is pretrained on roughly 36T tokens across 119 languages in three stages—a general stage (over 30T tokens at a 4,096 sequence length), a reasoning-oriented stage (∼5T higher-quality STEM, code, and synthetic tokens), and a long-context stage (hundreds of billions of tokens at a 32,768 sequence length)—followed by post-training that unifies a "thinking" and "non-thinking" mode under an adjustable thinking budget. **GPT-OSS-120B** [9] and **DeepSeek-R1** [10] are sparse mixture-of-experts (MoE) models that activate only a small fraction of their parameters per token (5.1B of 117B and 37B of 671B, respectively); DeepSeek-R1 in particular is built on the DeepSeek-V3 base and obtains its reasoning behavior through large-scale reinforcement learning with a cold-start supervised initialization. **Llama-4-Scout** [11] is a natively multimodal MoE model (17B active of 109B total, 16 experts, early-fusion vision) pretrained on roughly 40T tokens of text and image data with an exceptionally long 10M-token context window; it is the official task baseline and, unlike the other candidates, is not operated as an explicit reasoning model. **Claude Haiku 4.5** [12] is a proprietary model whose architecture and pretraining corpus are not publicly disclosed. These differences—dense vs. sparse activation, the presence or absence of an explicit reasoning mode, and open vs. closed weights—bear directly on both the accuracy/latency trade-offs in Table 1 and on the reproducibility discussion in Section 7.4.

4.2. Prompt Architecture

Our prompt augments a zero-shot LLM-as-judge baseline with three components: domain-matched few-shot examples, language-aware inline translation, and deterministic decoding. We describe each

below; the complete, verbatim prompt templates for both tracks are reproduced in Appendix A.

4.2.1. Domain-Matched Few-Shot Examples

For each evaluation instance, a single representative training sample is retrieved per class label, drawn from the corresponding domain as determined by matching the initial path component of the item's *fileid* attribute. In cases where no domain-specific correspondence can be established, the system defaults to general-purpose exemplars. Each retrieved sample comprises the question text, the associated answer text (truncated to a maximum of 300 characters), and the ground-truth class label.

This approach facilitates task-specific calibration by ensuring that each instance is evaluated relative to domain-congruent reference examples. For instance, a science question pertaining to photosynthesis is scored against other science-domain exemplars, rather than being compared to items drawn from unrelated domains such as history or literature.

4.2.2. Language-Aware Evaluation

Because rubrics are always written in English while student answers may appear in any of 13 languages, direct cross-lingual comparison risks surface-level pattern matching. Our prompt explicitly names the answer's language and instructs the model to translate it into English before evaluating:

```
The answer is written in {language_name}.
First, translate the answer to English.
Then, evaluate the translated answer against
the rubric criteria.
```

This forces the model to perform a comprehension step before scoring, ensuring that the comparison between answer and rubric occurs within a shared linguistic space.

4.2.3. Deterministic Decoding

We set temperature=0 (greedy decoding) for all inference, producing deterministic outputs for a fixed model endpoint and prompt. This reduces computational cost by a factor of 5× relative to a 5-run majority-voting scheme while keeping results consistent across repeated executions. We emphasize, however, that this is a cost-driven simplification rather than a substitute for multi-run evaluation: because Qwen3-32B is a reasoning model, the task guidelines recommend executing such models several times to estimate the variability of their outputs. Greedy decoding removes that variability by construction but consequently provides no estimate of the run-to-run uncertainty that sampling-based repetition would reveal (see Section 7.4).

4.3. Label Extraction

The model produces chain-of-thought reasoning followed by a final rating. We extract labels by:

1. Finding the last ASCII digit (0–9) in the response
2. Clamping to the valid range (0–2 for rubric, 0–4 for simple)
3. Defaulting to the middle label (1 or 2) if extraction fails

In practice, Qwen3-32B has an extraction failure rate of effectively 0%, making the fallback mechanism purely defensive.

4.4. Context Truncation

For items whose contexts exceed a substantial length threshold — primarily bilingual documents surpassing 25,000 characters — truncation is applied to ensure compatibility with the model's effective context window. This adjustment affects approximately 1% of all evaluated items. Given that the tokenization ratio for multilingual text is approximately 1.4 characters per token, the 25,000-character

truncation threshold corresponds to roughly 18,000 tokens, a figure that falls comfortably within the model's 32,000-token budget.

4.5. Simple Track Adaptation

For the simple rating track (0–4 scale), we adapt the approach with minimal modifications:

- The rubric descriptions are replaced with a generic 5-level quality scale (from "no relevant effort" at 0 to "clearly correct execution" at 4)
- Few-shot examples include one example per label (0–4) from the training set
- The label extraction range is adjusted to [0, 4]

All other components (domain matching, language awareness, translation instruction, deterministic decoding) remain identical.

5. Experimental Setup

5.1. Infrastructure and Reproducibility

All inference is performed through Amazon Bedrock [8] in the us-east-1 region using the Qwen3-32B endpoint. We issue 5–10 concurrent API requests to balance throughput against rate limits. No model fine-tuning, adapter training, or weight modification is performed; the system operates entirely through prompting. Because we use deterministic decoding (temperature=0), results are fully reproducible given the same model endpoint and prompt.

5.2. Data Splits and Evaluation Protocol

We use three data splits provided by the task organizers:

- **Training set**: used exclusively as the source for few-shot example retrieval. No training or optimization is performed on this data.
- **Development set**: 4,146 items with both rubric-based and simple track labels. We evaluate on the full set using the official scoring script to compute QWK and accuracy.
- **Test set**: approximately 93,500 items (46,752 rubric; 46,766 simple) scored in a single pass and submitted to the shared task leaderboard.

All hyperparameters—including the choice of one example per label, the 300-character truncation threshold for exemplars, and the 25K-character context limit—were fixed after initial experiments on a 50-item stratified sample (the same sample used for model selection in Section 4.1) and were not further tuned on the full development set. This reduces the risk of overfitting to the development distribution.

5.3. Evaluation Metrics

The primary metric is Quadratic Weighted Kappa (QWK) [6], as specified by the task organizers. We additionally report raw accuracy as a secondary diagnostic. For the ablation study, we compare QWK scores obtained by selectively disabling system components while keeping all other settings constant.

5.4. Computational Cost

Table 3 summarizes the inference cost and wall-clock runtime for each track. The total cost for both tracks across development and test sets is approximately $103, making the approach accessible without specialized hardware.

Table 3
Computational cost for full evaluation

Track	Items	Cost (USD)	Runtime (hours)
Rubric (dev)	4,146	$5	0.5
Simple (dev)	4,146	$5	0.5
Rubric (test)	46,752	$58	5
Simple (test)	46,766	$35	5

6. Results

6.1. Development Set Performance

Table 4
Development set results compared to the official baseline

Track	Our QWK	Baseline QWK	Improvement
Rubric-Based Rating	**0.821**	0.760	+0.061
Simple Rating	**0.607**	0.360	+0.247

Our system substantially outperforms the Llama-4-Scout-17B baseline on both tracks. The improvement is particularly pronounced on the simple track (+0.247 QWK), likely because the 5-level scale benefits more from few-shot calibration anchors than the 3-level rubric scale where explicit rubric descriptions already provide scoring guidance.

6.2. Ablation Study

To quantify the contribution of each system component, we conduct ablations on the full development set for the rubric track, disabling components individually and jointly:

Table 5
Ablation study on the rubric track (development set)

Configuration	QWK	Accuracy
Full system (few-shot + translation, temp=0)	**0.821**	**78.1%**
Without domain-matched few-shot	0.796	76.7%
Without language-aware translation	0.804	76.1%
Without both (zero-shot baseline)	0.805	76.7%

The ablation reveals a synergistic interaction rather than independent additive contributions. Neither component improves over the zero-shot baseline in isolation: translation alone slightly degrades QWK (0.796 vs. 0.805), while few-shot examples alone produce no meaningful gain (0.804 vs. 0.805). However, their combination yields a substantial improvement (+0.017 over the best single-component variant and +0.016 over the zero-shot baseline).

We hypothesize that this synergy arises because the two components address different failure modes that interact. Translation alone may introduce paraphrasing variation without providing scoring anchors, while few-shot examples alone offer English-language calibration that is less effective when compared against non-English answers. Their combination may allow translation to bring answers into the same linguistic space as the exemplars, while the exemplars provide scoring boundaries that stabilize the translated output.

Accuracy tells a partially different story than QWK: the full system achieves the highest accuracy (78.1%), and the QWK gains suggest that the combined approach reduces extreme errors ($|pred - ref| =$

2) that are quadratically penalized. This is consistent with the hypothesis that the synergy helps at the boundaries of the scoring scale.

6.3. Error Analysis

6.3.1. Rubric Track

On the rubric track, we observe:

- **Extreme errors** ($|pred - ref| = 2$): 38 out of 4,146 items (0.9%)
- **Primary failure mode**: Over-scoring answers that are topically relevant but factually incorrect (0→2 errors dominate)
- **Cross-lingual consistency**: QWK ranges from 0.788 to 0.843 across all 13 languages, indicating robust multilingual performance

6.4. Negative Results

We also explored several approaches that did not improve performance:

- **Structured output format** ("Rating: X"): Slightly hurts QWK by constraining reasoning
- **More few-shot examples** (2 per label): Adds noise and 5× slower, no improvement
- **Two-pass verification** (scorer + verifier): Only +0.002 QWK, not worth complexity
- **Ordinal decomposition** (binary cascade for simple track): Worse than holistic scoring
- **Language-specific prompt guidance**: Helps some languages but hurts others, net neutral

These negative results highlight that for this task, a simple but well-calibrated holistic scoring approach outperforms more complex multi-stage pipelines.

7. Discussion

7.1. Why Few-Shot Domain Matching Works

The training set contains items from the same educational domains as the test set. By providing domain-matched examples, we effectively show the model the "scoring culture" for that domain—how strictly or leniently similar items have been rated. This is analogous to how human raters undergo calibration training before scoring sessions.

7.2. Translation as a Reasoning Step

Instructing the model to translate before evaluating serves a dual purpose: (1) it ensures cross-lingual comparability, and (2) it forces an additional comprehension step that acts as implicit chain-of-thought reasoning [4]. The model cannot simply pattern-match against rubric keywords; it must first understand the answer deeply enough to translate it.

7.3. QWK-Aware Design Decisions

Several design choices are specifically motivated by QWK's quadratic penalty structure:

- **Conservative fallback**: When label extraction fails, we default to the middle label (1 for rubric, 2 for simple), minimizing expected quadratic loss.
- **Model selection**: We prioritized models with low extreme error rates over models with marginally higher accuracy but more 0↔2 confusions (as illustrated by DeepSeek R1's case).
- **Few-shot anchoring**: By showing all three label levels, we provide explicit boundaries that reduce the probability of skipping across adjacent categories.

7.4. Limitations

- **Reproducibility.** We access Qwen3-32B through the Amazon Bedrock API rather than self-hosted weights, so our exact results depend on the availability and behavior of that managed endpoint. However, Qwen3-32B is an *open-weights* model released under the Apache 2.0 license [7], and we report all generation parameters (greedy decoding, temperature=0, no repetition penalty or other logit modifications, the 32K-token context budget, and the prompt templates in Appendix A). The experiments should therefore be at least partially reproducible by anyone running the published weights with these settings—the principal caveat being that Bedrock may apply provider-side serving optimizations or proprietary fine-tuning that we cannot observe, which could cause minor divergence from a local Hugging Face deployment. To support exact replication against the public checkpoint, we note that the most recent revision of the `Qwen/Qwen3-32B` repository to modify the tokenizer configuration is commit `a09d5ccb54a84a572fcde9e54da24ec6e5a4b4a2` ("update tokenizer_config.json", 19 May 2025); we were unable to confirm the precise revision served by Bedrock and recommend pinning to this commit when reproducing our setup.
- **Single-pass, deterministic evaluation.** Because Qwen3-32B is a reasoning model, the task guidelines recommend running such models multiple times to characterize output variability. Our greedy, single-pass decoding eliminates run-to-run variance by construction but consequently provides no estimate of that variability; a multi-run sampling protocol (e.g., temperature$>$0 with majority voting and reported variance) would give a more complete picture at proportionally higher cost.
- The approach does not learn from errors—it uses static prompts without any form of active learning or feedback incorporation.
- Context truncation at 25K characters may lose relevant information for the 1% of very long items.
- The simple track's generic 5-level scale description was manually crafted and may not be optimal.

8. Conclusion

We presented a simple but effective approach to multilingual automated answer scoring for the CLEF 2026 ELOQUENT Sensemaking shared task. Our system combines domain-matched few-shot examples, language-aware inline translation, and deterministic decoding within a unified Qwen3-32B prompting framework. Without any model fine-tuning, we achieve QWK scores of 0.821 (rubric) and 0.607 (simple) on the development set, representing substantial improvements over the official baseline.

The key insight is that for multilingual scoring tasks, calibration (via domain-matched examples) and comprehension (via forced translation) are more important than model scale or complex multi-stage pipelines. Future work could explore dynamic example selection based on item difficulty, lightweight fine-tuning on the training set, or ensemble approaches combining multiple model perspectives.

Acknowledgments

We thank the ELOQUENT Sensemaking task organizers for providing the dataset and evaluation infrastructure, and Amazon Web Services for providing access to the Bedrock inference platform.

Declaration on Generative AI

During the preparation of this work, the authors used Qwen3-32B as the core scoring system described herein and Claude Opus 4.6 for grammar checking, language polishing, and an initial literature and citation search. We disclose that the generative-AI-assisted citation search initially surfaced one fabricated (hallucinated) reference—a non-existent "Overview of the CLEF 2025 shared task on sensemaking" attributed to incorrect authors—which we subsequently identified and removed. We have since manually verified every remaining reference against its primary source (publisher record, CEUR-WS proceedings,

arXiv, or official model documentation) to confirm that the authors, titles, venues, and identifiers are correct. The authors reviewed and edited all generated output and assume full responsibility for the accuracy and integrity of the final publication, including the correctness and existence of all cited works.

References

[1] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, Advances in neural information processing systems 36 (2023) 46595–46623.

[2] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-eval: Nlg evaluation using gpt-4 with better human alignment, in: Proceedings of the 2023 conference on empirical methods in natural language processing, 2023, pp. 2511–2522.

[3] P. Šindelář, O. Bojar, Overview of the sensemaking task at the ELOQUENT 2025 lab: LLMs as teachers, students and evaluators, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR-WS, 2025.

[4] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 1877–1901.

[6] J. Cohen, Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit, Psychological Bulletin 70 (1968) 213–220.

[7] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025).

[8] Amazon Web Services, Amazon Bedrock: Build and scale generative AI applications (2024). Available at https://aws.amazon.com/bedrock/.

[9] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao, et al., gpt-oss-120b & gpt-oss-20b model card, arXiv preprint arXiv:2508.10925 (2025).

[10] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, et al., Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, arXiv preprint arXiv:2501.12948 (2025).

[11] Meta AI, The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, 2025. Available at https://ai.meta.com/blog/llama-4-multimodal-intelligence/; model card at https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct.

[12] Anthropic, Introducing Claude Haiku 4.5, 2025. Available at https://www.anthropic.com/news/claude-haiku-4-5.

A. Full Prompts

This appendix provides the complete prompts used in our system for both tracks. Variables enclosed in braces (e.g., `{language_name}`) are populated at inference time.

A.1. Rubric Track

A.1.1. System Prompt

```
Please rate the given answer on a scale from 0 to 2
based on the rubric provided with respect to the
question and the information in the context. The answer
is written in {language_name}. First translate the
```

```
answer to English if needed, then evaluate its CONTENT
against the rubric. Think for a bit in english before
giving your final rating. Only put the argumentation
before the rating number, not after.

Here are examples of correctly rated answers:

--- Example (Rating: 0) ---
Question: {example_question}
Answer: {example_answer}
Rating: 0

--- Example (Rating: 1) ---
Question: {example_question}
Answer: {example_answer}
Rating: 1

--- Example (Rating: 2) ---
Question: {example_question}
Answer: {example_answer}
Rating: 2
```

A.1.2. User Prompt

```
Question:
{question}

Rubric:
0: {rubric_0_description}
1: {rubric_1_description}
2: {rubric_2_description}

Answer:
{answer}

Context:
{context}
```

A.2. Simple Track

A.2.1. System Prompt

```
Please rate the given answer on a scale from 0 to 4
with respect to the question and the information in
the context. The answer is written in {language_name}.
First translate the answer to English if needed, then
evaluate its CONTENT.

Rating scale:
0 No relevant effort or fundamental misunderstanding
  of the task
1 Minimal attempt to identify or apply relevant
  reasoning
2 Sound approach hindered by significant errors in
  key areas
3 Correct methodology with only minor slips or
  omissions
```

```
4 Clearly correct execution fully answering the
  question

Think for a bit in english before giving your final
rating. Only put the argumentation before the rating
number, not after.

Here are examples of correctly rated answers:

--- Example (Rating: 0) ---
Question: {example_question}
Answer: {example_answer}
Rating: 0

--- Example (Rating: 1) ---
...
--- Example (Rating: 4) ---
...
```

A.3. Few-Shot Example Selection

Examples are retrieved from the training set using domain matching: the first path component of each item's **fileid** determines its domain. One example per label is selected from the same domain; if no domain match exists, a generic fallback example is used. Question and answer text in examples are truncated to 300 characters.