

Al-Qahera at the FinEval 2026: GEPA Prompt Evolution for Greek Financial NER and Japanese Implicit Communication Recognition

Mohamed Mohamed
American University in Cairo
Cairo, Egypt
mabdelmagid@aucegypt.edu

Hazem Ajlan
American University in Cairo
Cairo, Egypt
hazemdiaz@aucegypt.edu

Abstract

We describe the Al-Qahera submission to two subtasks of the FinNLP 2026 Multilingual Financial Language Understanding Challenge: Greek financial numeric named-entity recognition (GRFinNUM) and Japanese implicit communication recognition (JF-ICR). For both subtasks we optimize a single frozen task model with GEPA, a reflective, natural-language prompt evolution method, using a smaller model as reflection critic. On the organizer’s own seed, GEPA raises held-out validation numeric-entity F1 from 0.473 to 0.947 (held-out test F1 0.978), and raises JF-ICR validation accuracy from 0.380 to 0.660 (held-out test accuracy 0.692), in both cases without any change to the task model itself; on the harder, like-for-like comparison against our own hand-written seed’s own held-out test F1 (0.961), the GEPA-optimized organizer prompt still comes out ahead (0.978). As a secondary exploration on Subtask 1, we also try Meta-Harness, a structurally different optimizer that searches over the surrounding program instead of the prompt; it is not part of our submitted predictions, but it surfaces a concrete case of search-set overfitting that we report as a finding in its own right.

1 Introduction

The FinNLP 2026 Multilingual Financial Language Understanding Challenge¹ poses three low-resource, non-English financial NLP subtasks. We participate in two: Subtask 1, Greek financial NER over Plutus-ben’s GRFinNUM/GRFinNER splits (Peng et al., 2025), and Subtask 2, five-way ordinal intent classification over EBISU’s JF-ICR investor-relations Q&A pairs (Peng et al., 2026). Both subtasks ship small labeled pools (a few hundred rows) and score a single frozen task model’s raw text output, which makes them a natural fit for *prompt* optimization rather than fine-tuning.

¹<https://www.finnlp2026sharedtask.info/>

Our system is built on the standalone gepa optimizer (Agrawal et al., 2025), via a hand-written adapter per subtask implementing its duck-typed GEPAAdapter protocol (§3): a task language model answers each example, and a separate, smaller reflection model rewrites the instruction text in response to natural-language feedback on the task model’s mistakes, iterating along a Pareto frontier of scored candidates. We run this same optimizer, unmodified, on both subtasks, despite their differing languages, label schemas, and modes of ambiguity, and report what actually changed in each seed instruction’s rewritten form (§4.1, §5). As a secondary exploration on Subtask 1 (§4.2), we also try Meta-Harness (Lee et al., 2026), which freezes the task model entirely and instead searches over the *code* that decides what to send it; we report it as an ablation against GEPA’s prompt-only search, not as part of our submitted predictions. The optimization scripts used to generate every reported number are released as part of our submission.

2 Data

Subtask 1 (GRFinNUM/GRFinNER). The labeled pool (320 train + 80 validation rows) covers only GRFinNUM, numeric entity extraction (types *chrimata* money, *pososta* percentage, *chronika* date/time, *posotites* quantity, *alla* other – Greek label strings, romanized here for typesetting), scored as exact-match set F1 over (entity text, type) pairs. We re-split it into 240 train / 80 validation / 80 held-out test examples, the last never touched by optimization. The 200-row test-participant file is half GRFinNUM and half GRFinNER (textual entities: person/organization/location), with zero labeled GRFinNER examples released; we answer that half zero-shot with the organizer’s own bundled prompt and report no held-out score for it, since we have no gold labels to score against.

Subtask 2 (JF-ICR). 253 labeled Japanese question/response pairs, each labeled with one of five ordinal intent levels (+2 strong commitment to −2 strong refusal), heavily imbalanced (+1: 139, 0: 72, +2: 29, −1: 11, −2: 2). We split it 151 train / 50 validation / 52 held-out test, scored by plain accuracy.

3 Shared Infrastructure

Both subtasks call Google AI Studio’s `gemma-4-31b-it` as the task model, round-robin across two API keys to double the effective rate limit, and `gemini-3.1-flash-lite` as the GEPA reflection model. Every example in a batch is dispatched concurrently via a thread pool; the GEPA engine itself evaluates one batch at a time and does not parallelize internally, so batch-level concurrency is implemented in our adapter. Each subtask defines a `GEPAAdapter` with two responsibilities: `evaluate()` runs the current instruction over a batch and scores it against gold, and `make_reflective_dataset()` turns scored trajectories into per-example natural-language feedback (score, gold output, predicted output) that the reflection model conditions on when proposing a revised instruction. GEPA search is stopped by whichever of two conditions fires first: no accepted improvement for a fixed number of iterations, or a cap on total candidate proposals.

4 Subtask 1: Greek Financial NER

4.1 GEPA Prompt Optimization

We run GEPA twice from different seed instructions, over the full 240/80 train/validation split: (a) a hand-improved seed with worked examples and an explicit output-format hint, and (b) the organizer’s own instruction text, verbatim, with no added examples. Table 1 reports validation and held-out test F1 for both; “Best Val F1” is the validation score of whichever candidate GEPA’s own Pareto selection judged best, which for the hand-improved run remained the seed itself.

The hand-improved seed starts far ahead on validation (0.955) of the organizer’s verbatim instruction (0.473), and GEPA’s search over the hand-improved run never proposes an accepted candidate that beats it within budget – so its 0.961 test F1 is the untouched seed’s own held-out performance, not an optimized result. Starting from the weaker organizer seed, GEPA instead closes most of that 0.482-point validation gap (0.473 to

Seed	Val F1	Best Val F1	Test F1
Hand-improved*	0.955	0.955	0.961
Organizer’s, verbatim	0.473	0.947	0.978

Table 1: GEPA prompt optimization on GRFinNUM (held-out test, 80 examples). The organizer-seed run explored 5 candidates over 6 iterations. *GEPA proposed no candidate that beat the hand-improved seed within budget, so this row’s Best Val F1/Test F1 are the unmodified seed’s own scores.

0.947) with an accepted rewrite, and the resulting instruction reaches a higher held-out test F1 (0.978) than the untouched hand-improved seed. We read this as a strong-seed-leaves-no-room-to-improve case next to a weak-seed-with-real-headroom case, rather than as two GEPA rewrites being compared directly – and, in this case, the optimized weak seed still ends up ahead of the untouched strong one. Our best-performing submitted instruction (the GEPA-optimized organizer seed, the third of five candidates explored) is used verbatim in §6.

4.2 Exploratory: Meta-Harness Program Search

GEPA optimizes a single prompt string around an otherwise fixed pipeline. As a secondary exploration, not part of our submitted predictions, we also try Meta-Harness (Lee et al., 2026), which instead freezes the *model* and searches over the *pipeline*: each candidate is a complete, standalone Python module implementing

```
predict(text, call_base_model,
train_examples) -> str
```

where `call_base_model` is the candidate’s only path to the (frozen) task model, and `train_examples` is the labeled pool available for retrieval or few-shot construction. A candidate may call the base model zero, one, or many times, in whatever pattern its own code implements.

We use `gemma-4-26b-a4b-it` (a 26B-total/4B-active MoE model) as the frozen base model, load-balanced across a four-key pool, and `gemini-3.1-flash-lite` as the proposer, which writes new candidates conditioned on the accumulated filesystem history of every prior candidate: its full source code, its aggregate score, and its worst-scoring execution traces. This browsable history, rather than a compressed summary, is what Lee et al. (2026) report is necessary for iteration to make progress, and it is what lets the proposer see, e.g., that a can-

Candidate	Strategy	Search F1	Held-out F1
iter6_cand1	retrieval FS	0.994	0.855
iter3_cand0	fixed FS	0.839	0.921
iter4_cand0	regex+classify	0.929	0.936

Table 2: Meta-Harness candidates on GRFinNUM (exploratory, not submitted). FS = few-shot. Search F1 is on the 24-example search set; held-out F1 is on a disjoint 12-example sample of the validation split.

candidate’s failures cluster on multi-part dates rather than percentages, and revise accordingly.

Each candidate is scored on a fixed 24-example sample of the train split (search set), never the validation split. We ran 10 iterations of 1–2 candidates each (18 candidates total), tracking a Pareto frontier over (score, average base-model calls per example). Table 2 reports the three candidates most relevant to our findings, each also evaluated once against a held-out 12-example sample of the validation split (a smaller, and separately drawn, held-out set than GEPA’s 80-example test split above, so the two systems’ numbers are not directly comparable example-for-example).

The top scorer on the search set, `iter6_cand1`, retrieves the two most textually similar `train_examples` for each input and uses them as few-shot context. It reaches near-perfect search F1 (0.994) but the worst held-out F1 of the three (0.855): with only 24 search examples, “most similar” retrieval can surface a near-duplicate of the query itself, a pattern that does not exist for genuinely new validation inputs. By contrast, `iter3_cand0` bakes in a single fixed few-shot example and never looks at `train_examples` at all; it scores lowest on the search set (0.839) but generalizes best among the two few-shot strategies (0.921). The best held-out candidate, `iter4_cand0`, sidesteps retrieval entirely: it extracts numeric-token candidates with a regular expression, then issues a single call asking the base model only to *classify* the pre-identified spans rather than to both find and classify them. This scores second-highest on the search set (0.929) and highest held-out (0.936), on one model call per example, the same cost as the other two strategies. Ranking candidates by held-out score rather than by the Pareto frontier’s own search-set ranking flips the winner entirely – the one piece of this pipeline that is not, and could not be, automated by the search itself – which is why we treat this as an exploratory finding rather than a candidate

	Val (50)	Test (52)
Seed (organizer’s header)	0.380	–
GEPA-optimized (13 cand.)	0.660	0.692

Table 3: GEPA on JF-ICR accuracy. Test is on the held-out 52-example split, never used for candidate selection.

Label	Support	P	R	F1
+2	3	0.000	0.000	0.000
+1	34	0.833	0.735	0.781
0	13	0.625	0.769	0.690
-1	1	0.167	1.000	0.286
-2	1	0.000	0.000	0.000

Table 4: Per-class precision/recall/F1 for the submitted JF-ICR instruction on the 52-example held-out test split (36/52 = 0.692 accuracy).

for submission: we did not have an equally-sized, independently-drawn held-out set to validate the choice on GEPA’s own footing (§4.1).

5 Subtask 2: Japanese Implicit Communication Recognition

We run a single GEPA optimization over the 151/50 train/validation split, seeded with the shared task’s own official prompt header verbatim (five ordinal labels, one-line definitions, “do not output these texts” instruction). `google_client.py` requests thinking tokens from the task model regardless of the seed instruction’s own “no explanation” wording, which we confirmed empirically keeps a genuine hidden reasoning channel open without leaking chain-of-thought into the scored answer. Table 3 reports the result.

GEPA’s accepted rewrites consistently add two ingredients the seed lacks: explicit criteria for distinguishing a polite, indirect refusal from a hedged commitment (a distinction Peng et al. (2026) identify as central to why this task is hard for general-purpose LLMs), and a small bank of Japanese qualifier phrases (e.g. *shinchō ni mite ori*, “viewing cautiously”) flagged as evidence for the +1 class specifically. The single largest error mode in every accepted candidate’s feedback was the +1/0 boundary, consistent with +1 being the majority class (139/253) and therefore the class whose ambiguity the accuracy metric weighs most heavily; later candidates’ rewrites increasingly focus their added rules on exactly that boundary rather than on the rarer classes.

Table 4 confirms the +1/0 boundary as the dominant error mode: of the 34 gold-+1 test examples,

5 are misclassified as 0, and of the 13 gold-0 examples, 3 are misclassified as +1 – together 8 of the 16 total test errors. The three minority classes (+2, -1, -2) have 1–3 test examples each, too few for a reliable per-class estimate; at that support, a single misclassified example moves F1 by 0.29–1.0, so we report these numbers for completeness rather than as evidence the optimized prompt handles the minority classes well or poorly.

6 Results Used for Submission

Our submitted predictions use, for Subtask 1, the organizer-seed GEPA instruction (Table 1, test F1 0.978 on the numeric half) run against the numeric half of the test-participant file, and the organizer’s bundled zero-shot prompt for the textual half; for Subtask 2, the GEPA-optimized instruction (Table 3, test accuracy 0.692). The Meta-Harness candidates of §4.2 were not submitted.

On the official FinEval 2026 leaderboard, our submission (team mabdelmagid) scores F1 0.9038 (precision 0.9167, recall 0.8912) on Subtask 1, ranked 5th, and accuracy 0.6571 on Subtask 2, ranked 10th. The Subtask 1 leaderboard score is computed over the full test-participant file (both the GRFinNUM numeric half we optimize and the GRFinNER textual half we answer zero-shot, §2), so it is not directly comparable to our numeric-only held-out test F1 (0.978, Table 1). The Subtask 2 leaderboard accuracy (0.6571) is close to, but not identical to, our own held-out test accuracy (0.692, Table 3), as expected from the two being scored on different (organizer vs. our own) held-out splits.

7 Discussion

GEPA’s gains on both subtasks come from the same mechanism – a smaller reflection model rewriting instruction text against natural-language feedback on scored trajectories, with no gradient update and no change to the task model itself – but the two subtasks stress that mechanism differently. Subtask 1’s feedback is almost entirely about output-format precision (span boundaries, entity-type labels, decimal-comma punctuation), so GEPA’s accepted rewrites concentrate on formatting rules. Subtask 2’s feedback is almost entirely about a semantic boundary between two adjacent ordinal classes, so its rewrites concentrate on distinguishing criteria and example phrases instead. That the GEPA-optimized organizer seed for Subtask 1 ends up ahead of our untouched hand-improved seed (Table 1) is a com-

parison between a rewritten weak prompt and an unrewritten strong one, not two rewrites – our hand-improved seed’s own validation score (0.955) was already high enough that GEPA’s search never produced an accepted candidate to test against it. The more direct lesson is the reverse of what a strong hand-written seed might suggest: leaving GEPA a real gap to close (the organizer seed’s 0.473) gave it room to find gains a stronger starting point did not need or permit.

GEPA’s Pareto selection already evaluates candidates against a validation set distinct from its own reflection minibatches, so it never faces the failure mode our Meta-Harness exploration ran into: with only a 24-example search set reused across the full proposer history, a retrieval-based candidate can quietly memorize its own evaluation set (iter6_cand1, Table 2) and rank first on search score while generalizing worst. That contrast is why we treated held-out score, not search score, as authoritative in §4.2, and why we limited GEPA itself to the submitted results in §6 rather than the exploratory harness search.

Does the prompt just get longer? Across both tasks, GEPA’s rewrites grow sharply past the seed but length does not otherwise track validation score. On GRFinNUM (organizer seed), the 354-character seed (val F1 0.473) is rewritten to 1665 characters at the first accepted candidate (val 0.686), but the best-scoring and submitted candidate (index 2, val 0.947) is shorter, at 1621 characters, and a later, shorter still candidate (index 3, 1178 characters) reaches val 0.940 – within 0.007 of the best candidate at 73% of its length. On JF-ICR, the submitted candidate (index 6, 2508 characters, val 0.660) is shorter than three later-explored candidates (3306–3542 characters) that all score lower on validation (0.44–0.52). GEPA’s proposer does try longer rewrites throughout the search, but the reflection process does not reward length on its own; the accepted gains come from which formatting rules or classification criteria a rewrite adds, not from how much text it adds.

Limitations

Both held-out test splits we report (80 examples for Subtask 1, 52 for Subtask 2) are small enough that single-point F1/accuracy differences of a few points should not be over-interpreted; we do not have variance estimates across seeds. The Meta-Harness held-out set (12 examples) is smaller still and was evaluated only once per candidate, so Ta-

ble 2’s held-out numbers are themselves subject to the same small-sample risk the harness search is trying to avoid on the search side; this is part of why we treat that exploration as a finding rather than a submission candidate. The Subtask 1 test-participant file is half GRFinNUM (numeric entities, which we evaluate and optimize, §4.1) and half GRFinNER (textual person/organization/location entities); we have no labeled GRFinNER data, so that half is answered zero-shot with the organizer’s own bundled prompt and carries no held-out score of any kind – the full submitted NER system is only partially evaluated. We do not report results for Subtask 3 (multilingual financial Q&A): our RLM-based exploratory pipeline did not reach a completed run before the submission deadline, and we chose not to submit partial or unvalidated numbers. Finally, both subtasks’ task and reflection models are accessed through a shared, rate-limited API tier; wall-clock and API-call budgets reported informally during development (not included here for space) were shaped as much by that infrastructure as by the search algorithms themselves, and may not transfer to a differently provisioned setup.

Acknowledgments

We thank the FinNLP 2026 organizers for releasing the shared task data and for the extension enabling this submission.

References

- Lakshya A Agrawal, Shangyin Tan, Dilara Soyulu, Noah Ziemis, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. 2025. GEPA: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*.
- Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. 2026. Meta-Harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*.
- Xueqing Peng, Triantafyllos Papadopoulos, Efstathia Soufleri, Polydoros Giannouris, Ruoyu Xiang, Yan Wang, Lingfei Qian, Jimin Huang, Qianqian Xie, and Sophia Ananiadou. 2025. Plutus: Benchmarking large language models in low-resource Greek finance. *arXiv preprint arXiv:2502.18772*.
- Xueqing Peng, Ruoyu Xiang, Fan Zhang, Mingzi Song, Mingyang Jiang, Yan Wang, Lingfei Qian, Taiki Hara, Yuqing Guo, Jimin Huang, Junichi Tsujii, and Sophia Ananiadou. 2026. Ebusu: Benchmarking large language models in Japanese finance. *arXiv preprint arXiv:2602.01479*.

A Submitted Instructions

For exact reproducibility, we reproduce the two instructions actually used to generate our submitted predictions (§6). Both are quoted directly from `candidates.json` in the released optimization run directories.

A.1 Subtask 1: Numeric-NER Instruction

The submitted instruction (organizer seed, GEPA-optimized, Table 1) is written in Greek; we quote it verbatim from `gepa_task1/runs/task1_organizer_full_run/candidates.json` (index 2), followed by an English translation for readers who do not read Greek.

Στόχος: Αναγνώριση και εξαγωγή αριθμητικών οντοτήτων από οικονομικές εκθέσεις.

Κανόνες Εξαγωγής: 1. Ατομικότητα: Αναλύστε τις οντότητες στα μικρότερα δυνατά αριθμητικά συστατικά. Ημερομηνίες: Διαχωρίστε τα αριθμητικά μέρη (π.χ. 31.12.2021 -> '31', '12', '2021') ως ΧΡΟΝΟΛΟΓΙΑ. Χρηματικά ποσά: Εξάγετε μόνο τον αριθμό (π.χ. '252.000') ως ΧΡΗΜΑΤΑ. Μην εξάγετε το σύμβολο του νομίσματος. Κωδικοί/Νόμοι: Διαχωρίστε τα αριθμητικά μέρη (π.χ. 3693/2008 -> '3693', '2008') ως ΑΛΛΑ. Ποσοστά: Εξάγετε τον αριθμό μαζί με το σύμβολο % (π.χ. '12%') ως ΠΟΣΟΣΤΑ.

2. Κατηγοριοποίηση: ΧΡΗΜΑΤΑ: Νομισματικές αξίες (μόνο αριθμοί). ΠΟΣΟΣΤΑ: Αριθμοί που συνοδεύονται από το σύμβολο %. ΧΡΟΝΟΛΟΓΙΑ: Έτη, ημέρες, μήνες (ως αριθμοί). ΠΟΣΟΣΤΑ: Αριθμητικά δεδομένα που μετρούν πλήθος αντικειμένων. ΑΛΛΑ: Αριθμοί που δεν εμπίπτουν στα παραπάνω (π.χ. αριθμοί νόμων, κωδικοί).

3. Ειδικοί Περιορισμοί & Διευκρινίσεις: Σύμβολα: Εξαιρέστε σύμβολα νομισμάτων (€), αλλά ΠΕΡΙΛΑΒΕΤΕ το σύμβολο % στις οντότητες ΠΟΣΟΣΤΑ. Διαχωρισμός: Εάν μια οντότητα αποτελείται από πολλαπλά τμήματα (π.χ. 3693/2008), διασπάστε τα σε μεμονωμένα στοιχεία, εκτός αν ο αριθμός αποτελεί ενιαία μαθηματική οντότητα (π.χ. 252.000). Αποκλεισμός: Μην εξάγετε αριθμούς που αποτελούν μέρος ονομάτων (π.χ. "Piraeus Port Plaza 1" -> το '1' δεν πρέπει να εξαχθεί ως ΑΛΛΑ, καθώς είναι μέρος ονομασίας και όχι αριθμητική οντότητα). Συνέπεια: Ένα έτος (π.χ. 2008) πρέπει πάντα να ταξινομείται ως ΧΡΟΝΟΛΟΓΙΑ, ακόμα και αν εμφανίζεται σε πλαίσιο αναφοράς νόμου.

Μορφή απάντησης: 'όνομα οντότητας, τύπος οντότητας' (μία ανά γραμμή).

English translation:

Goal: identify and extract numeric entities from financial reports.

Extraction rules: (1) Atomicity: break entities into their smallest possible numeric components. Dates: split numeric parts (e.g. 31.12.2021 → '31', '12', '2021') as *chronika*. Monetary amounts: extract only the number (e.g. '252.000') as *chrimata*; do not extract the currency symbol. Codes/laws: split numeric parts (e.g. 3693/2008 → '3693', '2008') as *alla*. Percentages: extract the number together with the % symbol (e.g. '12%') as *pososta*.

Categorization: *chrimata* = monetary values (numbers only); *pososta* = numbers with a % symbol; *chronika* = years, days, months (as numbers); *posotites* = numeric data counting a quantity of objects; *alla* = numbers not covered above (e.g. law numbers, codes).

Special constraints: exclude currency symbols (€), but include % in *pososta* entities. Split a multi-part entity (e.g. 3693/2008) into individual elements, unless the number is a single mathematical entity (e.g. 252.000). Do not extract numbers that are part of proper names (e.g. "Piraeus Port Plaza 1": the '1' is not *alla*, since it is part of a name). A year (e.g. 2008) is always *chronika*, even inside a law reference.

Answer format: 'entity text, entity type' (one per line).

would like to do X" is "+1", "we will do X" is a stronger commitment; a factual reason for negative news is "0", a promised recovery is "+1".

Examples of nuance: "we are considering share buybacks" → "+1" (not "0" or "+2"); "we cannot comment on M&A rumors" → "-1" (not "0"); explaining negative cash flow via seasonal factors → "0" (not "-1").

Do not output any explanation or reasoning. Output only the label.

A.2 Subtask 2: JF-ICR Instruction

The submitted instruction (GEPA-optimized, Table 3) is written in English and quoted here verbatim, from `gepa_task2/runs/task2_full_run/candidates.json` (index 6).

You are a Japanese financial expert specializing in IR (Investor Relations) and corporate communication. Your task is to analyze a financial question and the company's response to determine the "Intent Level" of the company regarding the subject matter.

Output exactly one label from: {"+2", "+1", "0", "-1", "-2"}.

Label definitions & nuance guidelines: "+2" (strong commitment): the company explicitly commits to a specific action, target, or outcome with high confidence. "+1" (weak/qualified commitment): the company expresses a positive intention or expectation but includes qualifiers (e.g. "we intend to," "we are considering," "we expect to," or mentions future conditions). "0" (neutral/hedged intent): the company provides factual explanations, status updates, or neutral analysis without committing to a specific future action or outcome; purely descriptive responses are "0". "-1" (weak refusal): the company politely declines to answer, deflects, or states it cannot provide information at this time. "-2" (strong refusal): the company explicitly rejects a premise or refuses to engage with the topic firmly.

Critical analysis strategy: distinguish "explanation" (no promised future result → "0") from "commitment" ("+"/"-2"); a polite Japanese refusal ("we cannot comment," "it is difficult to answer") is still "-1" even when the tone is soft; "we