OpenCódice at GRACE 2026: An Agentic Approach to Argument Mining on Spanish Clinical Texts

Francisco-Javier Rodrigo-Ginés[1,*], Jorge Chamorro-Padial[2]

[1]NLP & IR, Universidad Nacional de Educación a Distancia (UNED), 28040 Madrid, Spain
[2]OpenCódice Research, Spain

Abstract

We describe the OpenCódice submission to GRACE 2026 [1] at IberLEF, a two-track shared task on argument mining over Spanish clinical text. Track 1 asks systems to identify argumentative components (Premise, Claim, MajorClaim) and their directed relations (Support, Attack, Partial-Attack) on the abstracts of Spanish randomised controlled trials; Track 2 asks systems to detect relevant sentences, extract evidence spans, and link those spans to the correct answer of a multiple-choice clinical case from the Spanish medical-licensing examination. Both tracks are scored with strict character-level boundary matching between predicted and gold spans. The submission was produced inside a semi-autonomous research loop in which a coding agent built on Claude Opus 4.7 received the task description, the raw data, GPU access and API credentials, and iterated through a sequence of strategies under researcher supervision. A first fine-tuned XLM-RoBERTa pipeline reached development macro-F1 0.70 on Track 1 but collapsed to 0.006 on the blind test because the rule-based sentence splitter never aligned with gold entity offsets. Replacing the splitter recovered the alignment to 80.4% and lifted Track 1 to 0.2216, but roughly 20% of gold entities are sub-sentential clauses that no rule-based splitter recovers. The final system pivots from emitting character offsets directly to emitting verbatim text spans whose offsets are recovered by substring search. It is a GPT-5.4 zero-shot pipeline with structural constraints, which scored 0.2289 on Track 1 (ranked second out of five teams) and 0.5367 on Track 2 (ranked second out of three teams), for an overall GRACE score of 0.3828. The position defended in this paper is that agentic AI accelerates empirical NLP research, while hypothesis generation and the acceptance criterion remain with the human researcher.

Keywords

Argument mining, Clinical NLP, Strict-boundary evaluation, Zero-shot prompting, Agentic research, Spanish NLP

1. Introduction

GRACE 2026 [1] is the first Spanish-language shared task on argument mining in the clinical domain. The competition is organised around two independent tracks. Track 1, *Clinical Trial Evidence and Argumentation*, asks systems to identify argumentative components on the abstracts of Spanish randomised controlled trials (RCTs), with the component types Premise, Claim and MajorClaim, and to label the directed relations between component pairs as Support, Attack or Partial-Attack. Track 2, *Clinical Case Reasoning*, asks systems to mark relevant sentences in clinical cases drawn from the Spanish medical-licensing examination (MIR), to extract evidence spans inside those cases, and to link the spans to the correct answer of a five-way multiple-choice question. Both tracks are scored with strict character-level boundary matching between predicted and gold spans, with a relaxed token-level intersection-over-union metric reported as a diagnostic. The task is part of the IberLEF 2026 forum [2].

The paper documents a campaign run inside a semi-autonomous research loop, in the style of recent configurations [3, 4, 5]. A coding agent built on Claude Opus 4.7 [6] received the task description, the raw data, GPU access and API credentials, and handled the full implementation life cycle of every experiment: writing training scripts, launching them on remote GPUs, parsing logs, building submission files and revising working hypotheses in light of leaderboard returns. The researcher set the initial methodological framework at the start of the campaign, gated submissions and pivots, and otherwise stayed out of the operational execution of the loop. The central thesis of this work is that agentic

IberLEF 2026, September 2026, León, Spain
*Corresponding author.
frodrigo@invi.uned.es (Francisco-Javier Rodrigo-Ginés); jorge@opencodice.org (Jorge Chamorro-Padial)

AI accelerates empirical NLP research, while hypothesis generation, in the sense of deciding which research direction is worth pursuing, and the acceptance criterion, in the sense of deciding what counts as a successful result, remain with the human researcher.

The Track 1 trajectory illustrates that thesis. A fine-tuned XLM-RoBERTa [7] sequence classifier with a five-line regular-expression sentence splitter reached development macro-F1 0.70 on the component-identification subtask and 0.006 on the blind test partition. The cause was a splitter-to-gold offset misalignment: on the development partition, zero sentence chunks matched gold entity offsets exactly, because the regular expression greedily merged the abstract title (no terminator) with the first body sentence into a single 470-character chunk that the strict scorer could not match against any gold boundary. A replacement splitter that segments on terminal punctuation followed by uppercase letters, on runs of multiple whitespace, and on newlines lifted the splitter-to-gold alignment from approximately 0% to 80.4% and Track 1 to 0.2216 on a second submission. A structural ceiling remained: roughly 20% of gold entities are sub-sentential clauses that no rule-based splitter recovers, and no amount of class reweighting or minority oversampling solves a coverage problem that lies upstream of the classifier. The final system therefore pivots to a different output representation. Rather than emitting character offsets directly, the system asks a large language model (LLM) [8] to emit the verbatim text of each argumentative span and recovers the corresponding offsets by substring search against the source document. The pivot eliminates the splitter, the majority-overlap labelling, and the rare-class problem in one move. The submitted Track 1 system, GPT-5.4 zero-shot [9] with structural constraints, scored 0.2289 overall on Track 1, ranking second of five participating teams. The same prompting architecture, adapted with one task-specific rule (choice texts injected as Claim entities, so that the relation arguments are aligned with the scorer's expected ontology), scored 0.5367 on Track 2, ranking second of three. The overall GRACE score, the unweighted average of the two tracks, is 0.3828.

Roadmap. Section 2 positions the work in the literature on argument mining, Spanish biomedical NLP, and agentic research with language models. Section 3 describes the task formally, presents the splits and basic statistics of the GRACE dataset, and walks through the metric definitions. Section 4 documents the researcher-and-agent loop and the central pivot that defined the campaign. Section 5 describes the submitted system, subtask by subtask. Section 6 reports the official scores and the progression of submissions. Section 7 draws the transferable lessons, and Section 8 concludes.

2. Related Work

The work belongs to the intersection of three lines of research: argument mining, Spanish biomedical NLP, and agentic research with language models.

Argument mining has been an active line in NLP for over a decade, with persuasive essays serving as the canonical benchmark. The parsing-argumentation-structures work of Stab and Gurevych [10] on student essays established the joint task of component identification (Premise, Claim, MajorClaim) and relation detection (Support, Attack) on which subsequent literature is built, and the same component ontology is the one GRACE adopts on its RCT track. The transition from essay-style text to clinical text is non-trivial: the Mayer, Cabrio and Villata work on healthcare argument mining [11] adapted the same component scheme to English RCT abstracts and reported that the boundary-detection step is the bottleneck of the pipeline, even when the component-classification step is solved well. GRACE inherits that bottleneck. On a strict boundary metric, a system that classifies the right content but emits the wrong boundary is penalised exactly as a system that classifies the wrong content, and the rule-based segmentation step that is standard in the essay-mining literature does not transfer cleanly to clinical text, where titles, headings and parenthetical clauses break the sentence-as-unit assumption.

Spanish biomedical NLP is the second line. The available encoder options on Spanish text are dominated by XLM-RoBERTa [7] on the multilingual side, BETO [12] on the general-domain Spanish side, and `bsc-bio-ehr-es` [13] on the biomedical Spanish side. The biomedical encoder is, in principle, the right choice for the clinical-domain content of GRACE, but the dataset sizes (350 Track 1 training

Table 1
Track 1 training-set label distribution. Two severe imbalances are visible: MajorClaim is 2.8% of components and Attack is 2.5% of relations.

Type	Components	Relations
Premise	1 536	–
Claim	666	–
MajorClaim	64	–
Support	–	1 213
Partial-Attack	–	169
Attack	–	36

abstracts and 128 Track 2 training cases) are small relative to the parameter count of any of these encoders, and any fine-tuned system on this data is fundamentally bounded by the sentence-to-span coverage problem described above rather than by the encoder's biomedical pretraining. We report XLM-RoBERTa as the multilingual baseline and discuss in Section 7 why we believe a stronger biomedical pretraining would not have changed the campaign's outcome.

Agentic research with language models is the methodological frame for the campaign itself. The autoresearch framework of Karpathy [3], the AI Scientist of Lu and colleagues [4], the MLAgentBench evaluation of LLM agents on machine-learning experimentation [5], and the underlying ReAct loop [14] together describe a working configuration in which a language-model agent drives an end-to-end empirical pipeline. The configuration adopted here shares the per-experiment semi-autonomy of those systems but reserves hypothesis generation and the acceptance criterion for the human researcher.

3. Task, Data and Metrics

This section describes the two GRACE tracks in formal terms, presents the splits and basic statistics of the data, and lays out the official metrics. The description follows the task overview [1] and the data dictionaries provided to participants.

3.1. Track 1: Clinical Trial Evidence and Argumentation

The Track 1 data are the argument-annotated release of the Spanish AbstRCT Dataset [15] and ClinArgES dataset [16]. Throughout this paper we use *corpus* for the underlying collection of Spanish RCT abstracts and *dataset* for its argument-annotated release scored by the shared task.

Track 1 operates on abstracts of Spanish RCTs. Each abstract carries a list of `entities` (argumentative components with character-level `start`/`end` offsets and a type in {Premise, Claim, MajorClaim}) and a list of `relations` (directed pairs of entity identifiers with a type in {Support, Attack, Partial-Attack}). Two structural constraints define the legal relation space: Premise sources may target either a Premise or a Claim, and Claim sources may target either a Claim or a MajorClaim, with the latter restricted to Support and Attack only.

The training partition contains 350 abstracts, the development partition 60, and the blind test partition 389. Table 1 reports the label distribution on the training partition. Two severe imbalances stand out: MajorClaim is 2.8% of the components and Attack is 2.5% of the relations. Gold entity lengths have a median of 151 characters, with an inter-quartile range of 108 to 210 characters. Entities are typically sentence-level but not always: roughly one in five gold spans is a sub-sentential clause delimited by commas and a conjunction (for example, a span that begins after *"…, pero no …"*). The sub-sentential rate is the central data property of Track 1; we return to it in Section 4.

The official Track 1 score is the unweighted mean of two strict-boundary macro-F1 scores, one for component identification (Subtask 1.1) and one for relation detection (Subtask 1.2). A component prediction counts as a true positive only if its `(start, end, type)` triple matches a gold entity

exactly. A relation prediction counts as a true positive only if its source identifier, target identifier and relation type all match a gold relation exactly.

3.2. Track 2: Clinical Case Reasoning

Track 2 operates on clinical cases from the Spanish MIR examinations. Each instance is a triple of a clinical narrative, a clinical question, and a list of five multiple-choice answer options, exactly one of which is correct. The training partition contains 128 cases, the development partition 24 cases, and the blind test partition 102 cases. Annotations have three parts: a binary relevance label per sentence in the narrative, a list of evidence spans of type Premise or Claim inside the narrative, and a list of directed relations linking Premise spans to choice identifiers with a type in {Support, Attack}. The gold also emits Claim entities whose identifiers equal the choice identifiers $1 \ldots 5$, so that every relation has a Premise source and a Claim target inside the same identifier space.

Three subtasks are evaluated on Track 2. Subtask 2.1 (Evidence Sentence Detection) is binary sentence classification scored by F1 on the relevant class. Subtask 2.2 (Evidence Span Extraction) is span identification scored by strict micro-F1 with an end-to-end evaluation scope: predicted spans are evaluated against the full document, not against gold-relevant sentences. Subtask 2.3 (Relation Detection) is relation classification scored by strict macro-F1. The official Track 2 score is the unweighted mean of the three subtask scores.

3.3. Metrics: Strict Versus Relaxed Boundary Matching

The two boundary criteria are reported on every leaderboard line. The strict criterion requires `(start, end)` to match the gold offsets exactly; this is the official metric for the components, span and relation subtasks. The relaxed criterion requires only a token-level intersection-over-union (IoU) of at least 0.5 between the predicted span and a gold span of the same type; this is reported as a diagnostic, never as the official score.

The choice of strict matching on a clinical dataset interacts strongly with rule-based segmentation. Any system whose output boundary is determined by a sentence splitter inherits the splitter's alignment to the gold annotation, and any misalignment translates one-to-one into a true-positive count. A splitter whose alignment to gold is 80% caps the end-to-end strict score at 0.80 on the component subtask, and a splitter whose alignment is 0% produces a strict score of 0, regardless of how well the downstream classifier labels the content of each chunk. The structural consequence is that the correct architectural choice for a strict-boundary task is visible in the metric specification rather than in the model family.

4. Agentic Methodology

The campaign followed a two-role semi-autonomous loop. The agent is a Claude Opus 4.7 model [6] exposed to a working directory through a tool-use interface; it reads and edits files, executes shell commands, runs Python scripts on a remote GPU, queries OpenAI HTTP endpoints, and downloads artefacts from CodaBench. The human researcher seeds the loop with an initial set of working hypotheses (candidate model families, expected sources of discriminative signal, priors on which strategy classes are competitive), gates every submission, and authorises direction changes when a strategy hits a structural ceiling.

One iteration consists of four steps: (i) the agent proposes an experiment in natural language, (ii) the researcher approves or redirects the proposal, (iii) the agent codes, launches, and analyses the experiment, (iv) the agent returns a textual summary used to decide whether to submit, iterate, or pivot. Every tool call and reasoning step is recorded in a transcript log released with the code.[1] Figure 1 renders the loop graphically.

[1] Code, predictions and the reproduction recipe are available at https://github.com/franfj/GRACE.

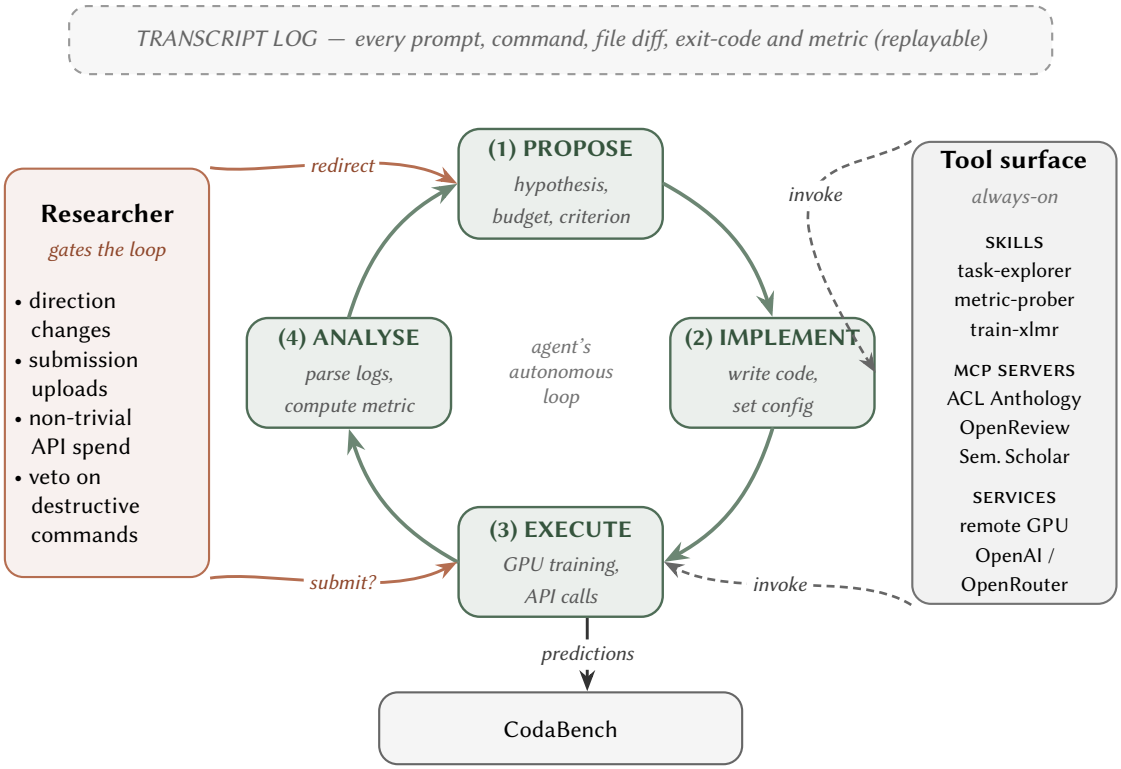

Figure 1: The semi-autonomous research loop. The agent proposes, codes, runs and analyses experiments; the researcher gates submissions and direction changes.

4.1. The Boundary-Matching Pivot

The Track 1 trajectory exhibits a single pivot that defined the campaign's outcome. The first system was a sentence-level pipeline: split each abstract into sentences, fine-tune an XLM-RoBERTa-base sequence classifier with four labels {None, Premise, Claim, MajorClaim} on the training-set sentences, assign training labels by majority overlap between sentence chunks and gold entity spans, and emit non-None sentences as entity predictions. The sentence splitter was a five-line regular expression of the form `r'[.!?]*[.!?]+|[.!?]+$'`, which captures every maximal substring terminated by one of `.!?` and the final trailing fragment. The classifier reached macro-F1 0.70 on the development partition and 0.006 on the strict-boundary blind test.

A splitter-to-gold alignment diagnostic on the development partition revealed the cause: of the 326 gold entities, zero had offsets exactly matched by any sentence chunk produced by the splitter. The regular expression `[.!?]*[.!?]+` greedily consumed the abstract title (no terminator) plus the first body sentence (terminated by `.`) into a single 470-character chunk that began at offset 0 and ended at offset 470, while the first gold entity in the abstract began at offset 1,147. The majority-overlap heuristic still assigned the chunk a Premise label because of the partial overlap with the nearest gold span. The model learned the correct content labels on the wrong spans, which the strict scorer recorded as zero true positives.

The replacement splitter, deployed in a second Track 1 submission, segments on three boundary classes: `.!?` followed by whitespace and an uppercase letter, runs of two or more whitespace characters, and newlines. Offsets are preserved in absolute coordinates against the source text rather than relative to any stripped copy. Splitter-to-gold alignment rose from approximately 0% to 80.4% on the development partition. Re-training the same XLM-RoBERTa classifier on the corrected chunks lifted the official Track 1.1 score to 0.3354 and Track 1.2 to 0.1078, for an overall Track 1 of 0.2216. The absolute score was capped by two structural properties of the data: roughly 20% of gold entities are sub-sentential clauses that no whitespace-and-punctuation rule recovers, and the minority classes (MajorClaim at 2.8% on the component side, Attack at 2.5% on the relation side) remain crushed by the majority class

even under a class-weighted loss.

The next move was a change of output representation rather than a refinement of the splitter. A model that emits `(start, end)` offsets directly must either match a sentence chunk or predict arbitrary offsets, and on 350 training abstracts the second option is hard to train. A model that emits the verbatim text of each argumentative span and recovers offsets via `raw_text.find(span_text)` sidesteps the splitter entirely: if the LLM copies a verbatim substring, the recovered offsets are by construction exact against the source document, and the only remaining constraint is that the emission must be a substring rather than a paraphrase. The pivot eliminates the splitter, the majority-overlap labelling, and the rare-class problem in one move. The new system was implemented as a single GPT-5.4 zero-shot call per abstract with a structured JSON output constraint and a post-processing step that locates each emitted span in the source text and rejects any entry whose text is not a verbatim substring. The same architecture was applied to Track 2 with one task-specific rule, described in Section 5.3.

5. System Description

We describe two systems: the fine-tuned XLM-RoBERTa baseline that produced the v1 and v3 submissions, and the GPT-5.4 zero-shot pipeline that produced the final submission on both tracks. The fine-tuned baseline is reported in detail because its trajectory motivates the pivot.

5.1. Fine-Tuned XLM-RoBERTa Baseline (Track 1)

The Track 1 baseline is a two-stage pipeline. Stage one is a sentence-level sequence classifier with four labels {None, Premise, Claim, MajorClaim}, trained on chunks produced by the rule-based splitter. The backbone is XLM-RoBERTa-base [7] with a tanh-activated linear classification head on the final-layer `[CLS]` representation. The loss is class-weighted cross-entropy with weights inversely proportional to per-class frequency on the training fold. Training uses AdamW [17] with learning rate 2×10^{-5}, batch size 16, five epochs, maximum sequence length 256 tokens, seed 42. Inference applies the trained classifier to every sentence chunk produced by the splitter on the test document, emits non-None chunks as predicted entities, and records their `(start, end)` from the chunk's absolute offsets.

Stage two is a directed-relation classifier over candidate component pairs. Each candidate pair is encoded as `(source_type: source_text [SEP] target_type: target_text)` and classified into {None, Support, Attack, Partial-Attack}. Negative (non-related) pairs are sampled from intra-document component pairs that do not appear in the gold relation list. Structural constraints are enforced at inference time as hard rules: a Premise source allows any of the three relation types to a Premise or Claim target; a Claim source allows Support or Attack only, to a Claim or MajorClaim target. The same XLM-RoBERTa backbone, the same optimiser settings, and a class-weighted cross-entropy loss are used.

Two iterations of the baseline were submitted. The v1 submission used the original five-line regular-expression splitter and scored Track 1 overall 0.006 on the blind test partition. The v3 submission used the corrected splitter and scored 0.2216. We describe in Section 6.3 how the gap between the two systems decomposes into a splitter-alignment recovery and a residual minority-class collapse.

5.2. GPT-5.4 Zero-Shot Pipeline (Track 1)

The submitted Track 1 system is a single per-abstract call to GPT-5.4 [9] with a structured JSON output constraint. The system prompt specifies the three component types with definitions, emphasises that MajorClaim is often absent in an abstract and should be labelled Claim under any uncertainty, describes the structural rules for relations (Premise sources can target a Premise or Claim with any of the three relation types; Claim sources can target a Claim or MajorClaim with Support or Attack only), and requires that the `text` field of every emitted entity be a verbatim substring of the abstract. The user message contains the abstract verbatim. The expected output is a JSON object with a `entities` list and a `relations` list, with the schema reproduced in Appendix C.3.

A post-processing step runs three checks before the prediction file is written. First, for each emitted entity, the `(start, end)` offsets are recomputed by `raw_text.find(entity.text)` on the source abstract, and any entity whose text is not a verbatim substring is discarded. Second, structural constraints on relations are enforced as hard rules: relations whose source or target identifier does not refer to an accepted entity are dropped, and relations whose source-target type combination is inconsistent with the constraints above are dropped. Third, a hard cap is applied to MajorClaim predictions: any MajorClaim beyond the first one in an abstract is demoted to Claim. The cap was set after preliminary dev runs showed repeated MajorClaim over-prediction on multi-paragraph abstracts.

5.3. GPT-5.4 Zero-Shot Pipeline (Track 2)

The Track 2 system is the same emit-text-then-locate architecture with two task-specific adaptations.

The first adaptation is sentence-relevance output. The user message includes not only the case narrative but also the ordered list of sentences with their indices, taken from the `metadata.context_sentences` block of the input record. The expected output extends the schema with a `sentence_relevancy` list whose length matches the input sentence list, with each element either `relevant` or `not-relevant`. The post-processing step verifies the length match and rejects predictions that are misaligned.

The second adaptation is the choice-as-Claim injection. The gold annotation places one Claim entity per multiple-choice option with identifier equal to the option identifier ($1 \ldots 5$). An earlier prompt variant asked the LLM to re-quote the choices as its own Claim entities and produced zero true positives on Subtask 2.3, because the recovered identifiers never matched the gold choice identifiers. The submitted variant skips the LLM on the Claim entities entirely: after the LLM returns its Premise spans, the post-processing step injects the five choices as Claim entities with `id = choice.id`, `text = choice.text` and `(start, end)` copied from the `metadata.choices` block. The injection is guaranteed to match the gold ontology by construction. Premise spans are subjected to the same verbatim-substring check as on Track 1. Relations are emitted with `arg1` pointing to a Premise span and `arg2` pointing to a choice identifier, with type in {Support, Attack}.

5.4. Negative Results We Did Not Submit

Two intermediate variants were generated and discarded before the final submission. We list them with their development behaviour, for the benefit of future participants.

- *XLM-RoBERTa with minority oversampling (Track 1).* A v4 baseline duplicated MajorClaim training examples by a factor of ten and Claim examples by a factor of two on the component side, and duplicated Attack by a factor of twenty and Partial-Attack by a factor of five on the relation side. Development macro-F1 improved from 0.683 to 0.687 on Subtask 1.1 and from 0.463 to 0.565 on Subtask 1.2 (a $+22\%$ relative jump on relations). The predictions were not submitted because the GPT-5.4 zero-shot pipeline produced higher development scores on the same partition in a comparable wall-clock budget. We expect a similar oversampling pattern would benefit any fine-tuned baseline on this data, and we record the absolute numbers above as a reference point.
- *Premise-only LLM emission for Track 2 (no choice injection).* An earlier Track 2 prompt asked the LLM to emit both Premise spans and Claim spans verbatim from the narrative, with choice texts excluded from the schema. The development micro-F1 on Subtask 2.2 was 0.14 and on Subtask 2.3 was 0, because the emitted Claim identifiers never matched the gold choice identifiers and every relation prediction failed the identifier check. Adding the choice injection lifted the same dev runs to micro-F1 0.65 on Subtask 2.2 and macro-F1 0.16 on Subtask 2.3.

6. Results

We report the results in three layers: the two official GRACE leaderboards (Tables 2 and 3), a per-subtask breakdown of the final GPT-5.4 submission (Table 4), and the progression of our three submissions on

Table 2
Official GRACE 2026 Track 1 leaderboard (Clinical Trial Evidence and Argumentation). Scores are macro-F1 percentages on the blind test partition. Best scores per column are in bold; our row is wrapped between rules. The baseline is reported on its own line at the bottom of the table.

Rank	Team	Overall	Subtask 1.1	Subtask 1.2
1	SINAI	**26.42**	**36.87**	**15.97**
2	OpenCódice	22.89	34.97	10.81
3	Verbanex AI	19.18	31.30	7.05
4	bionlp	18.42	36.83	0.00
5	UMUTeam	14.64	27.78	1.49
–	baseline (Encoder + Classifier)	3.48	6.83	0.12

Table 3
Official GRACE 2026 Track 2 leaderboard (Clinical Case Reasoning). Scores are F1 percentages on the blind test partition. Subtask 2.1 is F1 on the relevant class, Subtask 2.2 is strict micro-F1, Subtask 2.3 is strict macro-F1. Best scores per column are in bold; our row is wrapped between rules. The official baseline sits between our submission and the third-ranked team and is reported on its own line at the bottom.

Rank	Team	Overall	Sub. 2.1	Sub. 2.2	Sub. 2.3
1	VicomNLP	**59.84**	78.98	**72.55**	**27.99**
2	OpenCódice	53.67	**80.14**	64.75	16.13
3	UMUTeam	44.29	76.92	54.86	1.10
–	baseline (Encoder + Classifier)	47.90	73.48	62.63	7.58

Track 1 (Table 5). All scores are reported as percentages in the $[0, 100]$ range to match the leaderboard presentation; equivalent fractions in the $[0, 1]$ range are 0.2289 on Track 1, 0.5367 on Track 2 and 0.3828 overall.

6.1. Official Leaderboards

On Track 1, our submission is ranked second out of five teams, behind SINAI and ahead of Verbanex AI, bionlp and UMUTeam, and outperforms the official baseline on the overall score (22.89 against 3.48). The Track 1 field is dense at the top: the gap between rank one and rank two on the overall score is 3.53 percentage points, and the same gap on Subtask 1.1 is 1.90 points. On Track 2, the submission is ranked second out of three teams, behind VicomNLP and ahead of UMUTeam. We note that the official baseline exceeds the performance of one of the participating teams on the Track 2 overall score and sits between our submission (53.67) and the third-ranked team (44.29). On a per-subtask basis our submission outperforms the baseline on all three components of Track 2 ($+6.66$ on Subtask 2.1, $+2.12$ on Subtask 2.2, $+8.55$ on Subtask 2.3) and achieves the highest relevant-class F1 of all participating teams on Subtask 2.1 (80.14, ahead of VicomNLP's 78.98).

6.2. Per-Subtask Breakdown of the Final GPT-5.4 Submission

The relaxed-boundary scores on Subtasks 1.1, 1.2, 2.2 and 2.3 are consistently above the strict-boundary counterparts, with the largest absolute gap on Subtask 2.3 (28.98 relaxed against 16.13 strict). The interpretation is that the LLM's text emission is content-correct but boundary-approximate on a non-trivial fraction of spans: the model copies a verbatim substring that overlaps with a gold span but does not coincide with it, which the strict scorer penalises one-to-one. The gap is structural to the metric design rather than to the model, and any system that emits text rather than offsets faces the same ceiling on the strict criterion.

Table 4
Per-subtask breakdown of the submitted GPT-5.4 zero-shot system on the blind test partition. Strict scores are the official metric; relaxed scores (token-IoU ≥ 0.5) are reported as diagnostics. A dash indicates that the relaxed criterion is not defined for that subtask.

Subtask	Strict	Relaxed (IoU ≥ 0.5)
T1.1 Component identification	34.97	49.68
T1.2 Relation detection	10.81	19.21
T2.1 Evidence sentence (F1 relevant)	80.14	–
T2.2 Span extraction (micro)	64.75	74.16
T2.3 Relation detection (macro)	16.13	28.98
Track 1 overall	22.89	–
Track 2 overall	53.67	–
GRACE overall	38.28	–

Table 5
Progression of Track 1 submissions through the campaign. Scores are macro-F1 percentages on the blind test partition. The v1 system is the fine-tuned XLM-RoBERTa pipeline with the original five-line regular-expression splitter. The v3 system is the same pipeline with the corrected splitter described in Section 4.1. The final system is the GPT-5.4 zero-shot pipeline. A dash in the dev column indicates a run whose development score was not computed under the official scorer.

System	Dev (Sub. 1.1)	Test (Sub. 1.1)	Test (Sub. 1.2)
v1 XLM-RoBERTa, naive splitter	70.0	0.6	–
v3 XLM-RoBERTa, fixed splitter	68.3	33.54	10.78
Final GPT-5.4 zero-shot	–	34.97	10.81

6.3. Progression of Track 1 Submissions

Two regularities are visible. The first is that the dev/test gap on the v1 system was 69.4 percentage points on Subtask 1.1, large enough that no per-class tuning could explain it; the splitter-to-gold alignment diagnostic identified the cause in under a minute of agent wall-clock time. The second is that the corrected splitter recovered most of the content-classification signal that the v1 system had already learned: the v3 strict-test score of 33.54 is close to the v1 dev score of 70.0 scaled by the splitter alignment of 0.804, which yields 56.3, of which roughly 60% survives the residual minority-class collapse. The final GPT-5.4 system matches v3 on Subtask 1.1 and on Subtask 1.2 to within a percentage point, with the difference being that GPT-5.4 recovers the 20% of gold entities that are sub-sentential clauses, while v3 cannot, by construction. The two systems arrive at a similar overall score by different routes.

6.4. Where the Gap to the Leader Concentrates

The Track 1 gap to SINAI (-3.53 percentage points overall) decomposes into -1.90 on Subtask 1.1 and -5.16 on Subtask 1.2. On Subtask 1.1 the gap is narrow and likely reflects SINAI's domain-specific encoder pretraining (BETO with domain adaptation, per the leaderboard description) against our zero-shot prompting. On Subtask 1.2 the gap is wider and reflects the structural difficulty of relation detection under the strict-boundary criterion: a relation prediction is correct only if both arguments are correctly extracted and the relation type is correct, so the relation-detection error rate compounds the entity-detection error rate. The Track 2 gap to VicomNLP (-6.17 percentage points overall) decomposes into $+1.16$ on Subtask 2.1 (we lead), -7.80 on Subtask 2.2 and -11.86 on Subtask 2.3. The largest absolute gap is on Subtask 2.3, where the strict-boundary criterion combined with the rarity of Attack relations produces the same compounding effect as on Subtask 1.2.

7. Discussion

We identify three transferable lessons from the GRACE campaign that we expect to generalise to other strict-boundary span-annotation tasks.

The first lesson is that strict-boundary scoring and rule-based segmentation interact badly. A system whose output boundary is determined by a sentence splitter inherits the splitter's alignment to the gold annotation, and on a strict-boundary metric the inheritance is one-to-one. The v1 submission, with 0% splitter alignment, scored 0% on the strict metric despite a content-classification accuracy that the development set estimated at 70%. The v3 submission, with 80% splitter alignment, scored at the alignment ceiling. The recommendation for future participants on any strict-boundary task is to measure the splitter-to-gold alignment on the development set before any training run, and to treat the measurement as a hard cap on the achievable strict score.

The second lesson is that emit-text-then-locate is a structural fix for the strict-boundary problem. If a model emits the verbatim text of each span and recovers the offsets by substring search, the strict-boundary criterion is satisfied by construction whenever the emission is a verbatim substring. The remaining failure modes are (a) the LLM emits a near-paraphrase rather than a verbatim copy, in which case the substring search fails and the prediction is dropped, and (b) the LLM emits a verbatim substring that overlaps with a gold span but does not coincide with it, in which case the strict scorer counts a false positive. Both failure modes are controllable with prompting and post-processing, while the splitter-alignment failure mode is structural to rule-based segmentation and cannot be fixed by any change downstream of the segmenter. Our experience on Track 1 is that the pivot from one architecture to the other recovers a 20% sub-sentential-span coverage that no rule-based splitter reaches.

The third lesson is comparative. Our final GPT-5.4 submission is ranked second out of five teams on Track 1 and second out of three teams on Track 2, behind a single fine-tuned pipeline on each track (SINAI on Track 1, VicomNLP on Track 2). The submission achieves higher overall performance than the other fine-tuned approaches in the field, despite relying entirely on zero-shot prompting at inference time. The inference is not that prompting outperforms fine-tuning in general; the inference is that, when the training set is small (350 Track 1 abstracts and 128 Track 2 cases) and the metric is strict-boundary, the data efficiency of LLM zero-shot emission with substring recovery outperforms the boundary brittleness of a sentence-splitter-plus-classifier baseline. A domain-pretrained encoder may close the gap to the leaders on content classification, but it will not close the gap that lives upstream, in the sentence-segmentation step.

8. Conclusion

This paper described a semi-autonomous submission to GRACE at IberLEF 2026. The system, a GPT-5.4 zero-shot pipeline with structured JSON output and substring-recovery offset alignment, scored 0.2289 on Track 1 (ranked second out of five teams) and 0.5367 on Track 2 (ranked second out of three teams) on the blind test partition, for an overall GRACE score of 0.3828. The campaign was defined by one pivot: the move from emitting character offsets directly, through a fine-tuned XLM-RoBERTa pipeline whose strict-test score collapsed because the upstream sentence splitter never aligned with gold entity boundaries, to emitting verbatim text spans whose offsets are recovered by substring search. The pivot was authorised after the splitter-to-gold diagnostic exposed the structural ceiling of the fine-tuned baseline, and it lifted the submission to second place on both tracks. The campaign supports the position that agentic AI accelerates empirical NLP research, while hypothesis generation and the acceptance criterion remain with the human researcher. The code, predictions and reproduction recipe are released at https://github.com/franfj/GRACE.

Declaration on Generative AI

This submission used generative AI as part of the experimental loop and as the inference component of the final submitted system. Specifically, Claude Opus 4.7 (Anthropic) acted as a coding agent that semi-autonomously implemented experiments, ran them on remote GPUs, parsed logs and revised its working hypotheses in light of leaderboard feedback. Every tool call and every natural-language reasoning step taken by the agent was logged in the transcript file released with the code. GPT-5.4 (OpenAI) was used at inference time on the test partition as the underlying model of the final submitted pipeline (Track 1 and Track 2), with the prompts reproduced verbatim in Appendix C.

References

[1] I. De la Iglesia, A. Atutxa, A. Barrena, K. Gojenola, R. Martínez, S. Montalvo, M. Á. Rodríguez, S. Zakhir Puig, V. Gómez Martínez, J. Ramirez-Romero, J. M. Villa-Gonzalez, Overview of GRACE at IberLEF 2026: Granular Recognition of Argumentative Clinical Evidence, Procesamiento del Lenguaje Natural 77 (2026).

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

[3] A. Karpathy, Autonomous research with llm agents ("autoresearch"), https://karpathy.ai/, 2026. Open-source repository sketching an autonomous research loop driven by LLM agents.

[4] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, D. Ha, The AI scientist: Towards fully automated open-ended scientific discovery, arXiv:2408.06292, 2024.

[5] Q. Huang, J. Vora, P. Liang, J. Leskovec, MLAgentBench: Evaluating language agents on machine learning experimentation, in: Proceedings of the 41st International Conference on Machine Learning (ICML), 2024, pp. 20271–20309.

[6] Anthropic, Claude opus 4.7: System card, https://www.anthropic.com/news/claude-opus-4-7, 2026.

[7] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: Advances in Neural Information Processing Systems 33, NeurIPS 2020, 2020. URL: https://arxiv.org/abs/2005.14165.

[9] OpenAI, GPT-5.4 Thinking System Card, https://openai.com/index/gpt-5-4-thinking-system-card/, 2026.

[10] C. Stab, I. Gurevych, Parsing argumentation structures in persuasive essays, Computational Linguistics 43 (2017) 619–659.

[11] T. Mayer, E. Cabrio, S. Villata, Transformer-Based Argument Mining for Healthcare Applications, in: Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 2108–2115. doi:10.3233/FAIA200334.

[12] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish Pre-trained BERT Model and Evaluation Data, in: PML4DC at the 8th International Conference on Learning Representations (ICLR 2020), 2020.

[13] Barcelona Supercomputing Center, PlanTL, bsc-bio-ehr-es: A Spanish biomedical and clinical

RoBERTa encoder, https://huggingface.co/PlanTL-GOB-ES/bsc-bio-ehr-es, 2022. Spanish biomedical and clinical RoBERTa model released by the BSC PlanTL group.
[14] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, in: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023, 2023. URL: https://arxiv.org/abs/2210.03629.
[15] A. Yeginbergen, M. Oronoz, R. Agerri, Argument mining in data scarce settings: Cross-lingual transfer and few-shot techniques, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 11687–11699.
[16] S. Zakhir-Puig, V. Gómez-Martínez, M. Á. Rodríguez-García, S. Montalvo, R. Martínez, ClinArgES: Clinical argumentation spanish corpus, in: Computational Models of Argument - Proceedings of COMMA 2026, Frontiers in Artificial Intelligence and Applications, IOS Press, 2026.
[17] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019. URL: https://arxiv.org/abs/1711.05101.
[18] Anthropic, Introducing the Model Context Protocol, https://www.anthropic.com/news/model-context-protocol, 2024.
[19] I. Gusev, academia_mcp: Tools for automatic scientific research, https://github.com/IlyaGusev/academia_mcp, 2025. Model Context Protocol server exposing scientific-research tools, including the ACL Anthology bibliography.
[20] F.-J. Rodrigo-Ginés, J. Chamorro-Padial, openreview-mcp: A Model Context Protocol server for querying peer review at scale (v1.1), Zenodo, 2026. URL: https://doi.org/10.5281/zenodo.19772807. doi:10.5281/zenodo.19772807.
[21] J. Kuo, semanticscholar-MCP-Server: A Model Context Protocol server for the semantic scholar api, https://github.com/jackkuo666/semanticscholar-mcp-server, 2025. Model Context Protocol server exposing the Semantic Scholar academic graph.

A. Tool Surface: Skills

The campaign relied on a set of project-scoped skills written in the Claude Code skill format. Each skill is a markdown file with front matter and a body of natural-language guidance that the agent loads when it detects relevance. The skills used on this task are the same set that drove the rest of our IberLEF 2026 participation, listed below with the specific role they played on GRACE.

- `iberlef-task-explorer`: opens a fresh task by reading its description, downloading the data, and producing an exploratory report (label distributions, length distributions, train/test mismatch flags). On GRACE this skill flagged the heavy minority-class imbalance on both component and relation labels (MajorClaim at 2.8%, Attack at 2.5%) and the sub-sentential nature of roughly 20% of gold entity spans on Track 1.
- `iberlef-metric-prober`: characterises the official scoring metric empirically before optimising. On GRACE this skill identified the strict character-level boundary criterion as the dominant property of the metric and recommended a splitter-to-gold alignment check before any fine-tuning iteration.
- `iberlef-baseline`: produces a TF-IDF baseline, a fine-tuned XLM-RoBERTa-base baseline, and an LLM zero-shot prompt. On GRACE this skill produced the v1 and v3 XLM-RoBERTa baselines on Track 1 and the GPT-5.4 zero-shot system that became the final submission.
- `iberlef-train-xlmr`: a parameterised fine-tuning skill that wraps XLM-RoBERTa-base under a common training recipe with class-weighted cross-entropy, optional minority oversampling, and early stopping on the development score.
- `iberlef-codabench-submit`: packages a prediction file in the format the task expects, runs format validation, computes a local score for sanity, and prepares the archive for the researcher to upload.

Table 6
Hyperparameters of the two submitted systems. The fine-tuned baseline (v1/v3) produced the XLM-RoBERTa Track 1 runs; the GPT-5.4 zero-shot pipeline produced the final Track 1 and Track 2 submissions.

Setting	Value
Fine-tuned XLM-RoBERTa baseline	
Backbone	XLM-RoBERTa-base
Optimiser	AdamW
Learning rate	2×10^{-5}
Batch size	16
Epochs	5
Max sequence length	256 tokens
Loss	Class-weighted cross-entropy
Random seed	42
GPT-5.4 zero-shot pipeline	
Model	`gpt-5.4`
Decoding	Default (zero-shot, no exemplars)
Output format	JSON object (entities, relations)
Max completion tokens	3,000
Retries per instance	up to 3
Offset recovery	`raw_text.find(span_text)`
Post-filter	Verbatim-substring check; structural relation rules; one-MajorClaim cap

B. Tool Surface: MCP Servers

Three Model Context Protocol servers [18] were active during the campaign. The `academia_mcp` server [19] exposed the ACL bibliography as a queryable resource, used by the agent to retrieve prior work on argument mining and on RCT-domain component annotation. The `openreview-mcp` server [20] extended the same capability to NeurIPS and ICLR; on this task the agent used it to ground the semi-autonomous research and ReAct-style citations in Section 2. The `semanticscholar-mcp-server` [21] provided forward and backward citation expansion, used to identify the foundational argument-mining literature (Stab and Gurevych, Mayer and colleagues) cited in Section 2.

C. Reproducibility: Hyperparameters, Prompts and Example I/O

This appendix collects the information needed to reproduce the submitted systems without consulting the external repository: the hyperparameters of both the fine-tuned baseline and the GPT-5.4 inference pipeline (Table 6), a worked input/output example in the exact JSON schema the system consumes and emits, and the GPT-5.4 system prompts reproduced verbatim. The user message in every case is the verbatim source text of the abstract or clinical case. The expected output is a JSON object whose schema matches the GRACE data format described in Section 3.

C.1. Hyperparameters

C.2. Example Input and Output (Track 1)

The listing below shows a compact Track 1 instance in the input schema (an abstract with its identifier) and the corresponding GPT-5.4 output after post-processing, with the recovered character offsets. The `text` field of every entity is a verbatim substring of `raw_text`, which is what makes the `(start, end)` recovery exact under the strict scorer.

```
// INPUT (one Track 1 record)
{
  "id": "RCT_0421",
```

```
  "raw_text": "El tratamiento redujo la presion arterial de forma significativa. No se observaron
      ↪ efectos adversos graves."
}

// OUTPUT (after verbatim-substring post-processing)
{
  "id": "RCT_0421",
  "entities": [
    {"id": "T1", "type": "Premise", "start": 0, "end": 60,
     "text": "El tratamiento redujo la presion arterial de forma significativa."},
    {"id": "T2", "type": "Claim", "start": 61, "end": 106,
     "text": "No se observaron efectos adversos graves."}
  ],
  "relations": [
    {"id": "R1", "arg1_id": "T1", "arg2_id": "T2",
     "relation_type": "Support"}
  ]
}
```

C.3. Track 1 System Prompt

```
You are an expert annotator of argument mining on Spanish randomised controlled trial abstracts (
    ↪ GRACE 2026, Track 1).
Identify the argumentative components and the directed relations between them.

Component types:
- MajorClaim: the central, abstract-level argumentative thesis. Often the title or the conclusion
    ↪ sentence. At most one MajorClaim per abstract; if uncertain, label as Claim.
- Claim: a specific argumentative assertion supported (or attacked) by evidence in the abstract.
- Premise: a piece of evidence (result, observation, statistic) that supports or attacks a Claim
    ↪ or another Premise.

Relation types and structural constraints:
- Premise -> Claim or Premise: Support, Attack or Partial-Attack.
- Claim -> Claim or MajorClaim: Support or Attack only.
- No other source-target type combination is legal.

Output a JSON object with two fields:
- "entities": list of {id, text, type}, where text is a VERBATIM substring of the abstract.
- "relations": list of {id, arg1_id, arg2_id, relation_type}, where arg1_id and arg2_id are entity
    ↪  ids defined above.

CRITICAL: the "text" field of every entity must be a verbatim substring of the abstract. Do not
    ↪ paraphrase, do not normalise punctuation, do not insert ellipses.
```

C.4. Track 2 System Prompt

```
You are an expert annotator of evidence-based reasoning on Spanish MIR (medical-licensing)
    ↪ clinical cases (GRACE 2026, Track 2).
You are given a clinical narrative, a clinical question, and a list of five answer options (one
    ↪ correct).

Your job has three parts:

1. Mark each sentence in the narrative as "relevant" or "not-relevant" to the clinical question.
    ↪ Return the labels in an ordered list matching the indices of the input sentences.

2. Extract Premise spans from the narrative. A Premise is a piece of clinical evidence (sign,
    ↪ symptom, finding, history) that bears on the question. Each Premise must be a VERBATIM
    ↪ substring of the narrative.
```

```
3. Link each Premise to one or more answer options with a relation type of Support or Attack. Use
    ↪ the option id (1, 2, 3, 4 or 5) as the relation target. Do NOT emit Claim entities
    ↪ yourself; the choices are injected externally as Claims with id equal to the option id.

Output a JSON object with three fields:
- "sentence_relevancy": list of strings ("relevant" or "not-relevant"), one per sentence in the
    ↪ input order.
- "entities": list of {id, text, type=Premise}, with text a verbatim substring of the narrative.
- "relations": list of {id, arg1_id (a Premise id), arg2_id (a choice id 1-5), relation_type ("
    ↪ Support" or "Attack")}.

CRITICAL: every Premise "text" must be a verbatim substring of the narrative. Do not paraphrase.
```