SINAI-UGPLN at CheckThat! 2026: Confidence-Aware Reasoning Trace Ranking for Numerical Claim Verification

CheckThat! Lab at CLEF 2026

Mariuxi del Carmen Toapanta-Bernabé[1,2,*,†], Miguel Ángel García-Cumbreras[1,†] and Luis Alfonso Ureña-López[1,†]

[1]Computer Science Department, SINAI, CEATIC, Universidad de Jaén, 23071, Jaén, Spain
[2]Universidad de Guayaquil, 090514, Guayas, Ecuador

Abstract

This working note describes the SINAI-UGPLN team's participation in CheckThat! 2026 Task 2, *Fact-Checking Numerical Claims*. The task requires systems to verify numerical claims by jointly ranking LLM-generated reasoning traces and selecting a final veracity verdict from evidence-supported reasoning paths. We approached the task as a combined trace-ranking and verdict-selection problem for English and Spanish. Rather than treating the reasoning traces as independent explanations, our systems used them as competing verification signals whose usefulness depends on numerical consistency, temporal alignment, verdict agreement, and confidence. We explored two main families of methods: supervised thresholded verification based on TF-IDF and Logistic Regression, and confidence-aware LLM-assisted verification over uncertain examples. The final English submission used Mistral as a non-fine-tuned verifier for selected uncertain cases and achieved an official score of 0.395. The final Spanish submission used supervised training and validation thresholding and achieved an official score of 0.458. Our results show different behavior across languages: LLM-assisted verification provided a small improvement for English, whereas Spanish was better served by supervised thresholding. This suggests that numerical claim verification with reasoning traces requires language-specific calibration and that LLM-based post-processing should be applied with caution when not fine-tuned to the target language and label distribution.

Keywords

numerical claim verification, fact-checking, reasoning trace ranking, test-time scaling, confidence-aware verification, multilingual NLP, CheckThat! 2026

1. Introduction

Numerical claims are a recurring component of public misinformation because they often appear precise, objective, and easy to quote. Claims involving percentages, budgets, dates, rankings, demographic figures, economic indicators, or temporal comparisons can be persuasive even when they are misleading. Their verification is difficult because the decisive evidence may depend on a small numerical difference, a specific time window, a denominator, or the interpretation of a comparison. A claim may be topically related to a piece of evidence while still being false because the quantity, date, or scope does not match. For this reason, numerical claim verification requires more than topical retrieval: it also requires reasoning about whether the evidence supports, contradicts, or leaves the numerical relation expressed in the claim unresolved.

CheckThat! 2026 Task 2 addresses this problem through a test-time scaling setting for numerical fact-checking [1, 2]. For each claim, the task provides evidence and multiple reasoning traces generated by a large language model (LLM). Each trace is associated with a candidate verdict, and participating systems must rank the traces by usefulness and produce a final veracity label. This formulation reflects a

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally.
mctb0005@red.ujaen.es,mariuxi.toapantab@ug.edu.ec (M. d. C. Toapanta-Bernabé); magc@ujaen.es (M. : García-Cumbreras); laurena@ujaen.es (L. A. Ureña-López)
0000-0002-4839-7452 (M. d. C. Toapanta-Bernabé); 0000-0003-1867-9587 (M. : García-Cumbreras); 0000-0001-7540-4059 (L. A. Ureña-López)

practical challenge in LLM-assisted fact-checking pipelines: generating multiple rationales can increase coverage and diversity, but it also creates the need for a verifier that can identify which reasoning paths are reliable [3, 2].

This setting differs from source retrieval tasks, where the primary output is a ranked list of documents. In Task 2, the system must operate on existing evidence and reasoning traces, combining a ranking problem with a classification problem. The trace ranking component determines which reasoning paths should be trusted, while the verdict-selection component determines the final label from the available signals. Consequently, the task can be understood as a joint reasoning-trace ranking and verdict-selection problem. This interpretation guided our participation in the English and Spanish tracks.

The SINAI-UGPLN team participated in English and Spanish. We did not submit a system for Arabic. Our approach explored language-specific configurations grouped into two main methodological families. The first family used supervised classical verification based on TF-IDF representations, Logistic Regression, and thresholded decision rules. This family was particularly effective for Spanish when trained using training and validation data. The second family used confidence-aware LLM-assisted verification, in which Mistral served as a non-fine-tuned verifier on selected uncertain English examples. In this setting, the LLM was not treated as a trained model; instead, it served as a controlled judge whose output was accepted only subject to confidence constraints.

The final systems selected for submission were language-specific. For English, the best confirmed configuration was a confidence-aware LLM-assisted verifier, which applied confidence-aware LLM-assisted post-processing to uncertain examples and achieved an official score of 0.395. For Spanish, the best confirmed configuration was the supervised thresholded verifier, which used supervised training and validation thresholding and achieved an official score of 0.458. The fact that the best English and Spanish systems came from different families is central to our analysis. It indicates that an LLM-assisted verifier may help selected English cases but does not necessarily transfer to Spanish without additional calibration or fine-tuning.

The contributions of this working note are fourfold. First, we formulate Task 2 as a joint reasoning-trace ranking and verdict-selection problem rather than as a simple final-label classification task. Second, we describe a supervised thresholded verifier for Spanish based on training and validation data and calibrated decision rules. Third, we report a confidence-aware LLM-assisted verifier for English uncertain cases, explicitly distinguishing it from a trained neural verifier. Fourth, we provide an integrated comparison of English and Spanish official results, including non-final variants only as supporting analysis and without inferring unavailable metric breakdowns.

The rest of the paper is organized as follows. Section 2 describes the official task, input format, output format, and evaluation setting. Section 3 motivates numerical claim verification, reasoning traces, and verifier-based ranking. Section 4 presents the system architecture. Section 5 details the experimental setup and variant families. Section 6 reports the official English and Spanish results. Sections 7, 8, and 9 discuss the findings, error sources, and limitations, and Section 10 concludes the working note.

2. Task Description

CheckThat! 2026 Task 2, *Fact-Checking Numerical Claims*, is defined as a test-time scaling task for fact verification within the CheckThat! 2026 multilingual verification pipeline [1]. For each claim, the organizers provide relevant evidence and multiple reasoning traces, each with a corresponding candidate verdict. Participants must rank the reasoning traces according to their usefulness for reaching the correct verdict and output a final verdict from the top-ranked or highest-quality reasoning signals [1]. Thus, the task combines two complementary objectives: trace ranking and verdict classification.

The official task covers English, Spanish, and Arabic. The dataset statistics reported by the organizers are shown in Table 1. Although the task includes three languages, the SINAI-UGPLN participation reported in this working note focused on English and Spanish. We did not submit an Arabic system.

Each example consists of a numerical claim, associated evidence, multiple LLM-generated reasoning

Table 1
Official Task 2 dataset statistics reported by the organizers.

Language	Number of claims	Used by SINAI-UGPLN
English	8000	Yes
Spanish	2808	Yes
Arabic	3260	No

traces, and a candidate verdict for each trace. The system must produce two outputs: a final veracity verdict and a ranking of the reasoning traces by their usefulness for reaching that verdict. The official verdict labels are `False`, `True`, and `Conflicting`. This structure makes the task distinct from standard claim classification, as the system must decide both which reasoning paths to trust and which final label to assign.

The official evaluation combines ranking and classification. Ranking quality is assessed using Recall@5 and MRR@5 over the reasoning traces, while verdict classification is assessed using macro-F1 and classwise F1. The official leaderboard score was used as the final reported score in this working note. Since the official per-metric breakdown was not available for all submitted variants, we do not infer macro-F1, Recall@5, MRR@5, or classwise F1 from the official aggregate score.

We validated each submission against the official task constraints before upload [4]. The validation checked the number of predictions, the use of valid verdict labels, and the consistency between the number of candidate verdicts, reasoning traces, and trace-ranking scores for each claim.

3. Background and Motivation

Numerical claim verification is a specialized form of fact-checking in which the decisive evidence often depends on a quantity, date, rate, proportion, ranking, or temporal comparison [5, 2]. A claim can be topically related to a source while still being incorrect because the number is outdated, the denominator differs, the unit has changed, or the statement compares incompatible time periods. This makes numerical verification more fragile than general topical matching. The system must identify not only whether the evidence is relevant, but also whether it resolves the specific numerical relation expressed by the claim.

This difficulty is particularly visible in claims involving percentages, budgets, inflation rates, employment figures, public expenditure, demographic indicators, or policy effects. For example, a reasoning trace may mention the same institution or event as the claim but use a different fiscal year; another trace may correctly identify a percentage but fail to determine whether it refers to an increase, a decrease, an accumulated total, or a year-on-year comparison. In such cases, the trace may appear plausible while still leading to an incorrect verdict. Therefore, ranking reasoning traces requires sensitivity to numerical alignment, temporal scope, and evidential sufficiency.

The test-time scaling setting used in Task 2 introduces multiple LLM-generated reasoning traces for the same claim. This design is useful because different traces may explore different aspects of the evidence. However, it also creates a verifier problem. Some traces may be redundant, some may be partially correct, some may contradict each other, and some may reach the right verdict with weak or incomplete reasoning. The task is therefore not simply to accept the majority verdict among the traces, but to identify which traces provide the most reliable basis for final decision-making.

This setting is related to Best-of-N and self-consistency approaches, where several model outputs are generated and then aggregated or selected [3, 2]. In numerical fact-checking, however, naive aggregation can be risky. A majority of traces may repeat the same numerical misunderstanding, while a minority of traces may correctly identify the decisive mismatch. Conversely, a single trace with high confidence may over-correct the prediction if it misinterprets the evidence. For this reason, the verifier must balance trace ranking, verdict aggregation, and confidence-aware selection.

From a system-design perspective, Task 2 can be framed as a joint trace-ranking and verdict-selection problem. The ranking component estimates which reasoning traces are most useful, while the verdict-

selection component determines the final label from the ranked traces, model probabilities, and confidence signals. In our participation, this framing motivated two language-specific approaches. For Spanish, we used a supervised classical verifier based on TF-IDF, Logistic Regression, and thresholded decision rules, trained on the training and validation data. For English, we used a confidence-aware LLM-assisted verifier over selected uncertain examples, where Mistral acted as a non-fine-tuned judge rather than as a trained classifier [6, 7].

The contrast between English and Spanish is central to this working note. The same LLM-assisted strategy did not transfer equally across languages. English obtained its best confirmed score with confidence-aware LLM-assisted post-processing, whereas Spanish obtained its best confirmed score with supervised training and validation thresholding. This suggests that multilingual numerical claim verification requires language-specific calibration and that LLM-based verification should be applied with caution when the model is not fine-tuned to the target language, evidence style, and label distribution.

Our previous participation in CheckThat! motivated the use of multilingual and evidence-aware verification strategies. Nevertheless, the present working note reports only Task 2 experiments. Task 1 source retrieval is a separate task and is not part of the methodology described here.

4. System Architecture

The architecture used by SINAI-UGPLN for Task 2 receives a claim, associated evidence, a set of LLM-generated reasoning traces, and one candidate verdict per trace. The output preserves the official structure by providing a final verdict together with trace-ranking scores. The verification process is organized into four stages: input normalization, trace scoring, verdict aggregation, and output validation.

4.1. Input Representation

Each official example was represented as a structured verification instance composed of the claim text, the available evidence snippets, the list of Best-of-N reasoning traces, and the verdict assigned to each trace. We normalized all verdict labels into the official label set: `False`, `True`, and `Conflicting`. This normalization was necessary because different components of the pipeline used lowercase or internal representations during processing, while the final JSON output required the official label format.

The reasoning traces were treated as competing verification signals rather than as independent explanations. For each example, the system preserved the original correspondence between the candidate-verdict list, reasoning traces, and trace-ranking scores. This constraint was central to the implementation: a valid prediction required that each trace had exactly one candidate verdict and exactly one score. Any mismatch in these lengths would produce an invalid submission or an incorrect trace ranking.

The claim-level representation used by the supervised variants combined the claim text, available evidence, partial reasoning-trace information, and aggregate features derived from the candidate verdict distribution. These aggregate features encoded whether the reasoning traces tended to agree on a verdict or whether they presented conflicting signals. The motivation was that a claim with mixed trace verdicts should be treated differently from one where most traces converge on the same label.

4.2. Trace Scoring and Ranking

The trace-ranking scores determine how reasoning traces are ranked for evaluation. We explored several trace-scoring mechanisms. Some variants preserved or lightly adjusted the base trace ranking. Others replaced trace scores using classifier probabilities, verdict agreement, numerical cues, or LLM-assisted scores. The objective was not only to produce a final verdict but also to assign higher scores to traces that were more useful for reaching that verdict.

The trace-ranking component was especially important because Task 2 evaluates the usefulness of reasoning traces, not only the final label. A trace can be topically related to the claim but still fail to resolve the decisive numerical or temporal comparison. Therefore, trace scoring should favor traces

that explicitly address the relevant quantity, date, comparison, or evidential contradiction. In practice, our variants approximated this behavior through class-probability scoring, majority-verdict bonuses, numerical cue weighting, and LLM-provided trace scores.

Ranking-only variants modified the trace-ranking scores while preserving the final verdict. These variants tested whether improving trace ordering alone was sufficient. Full-replacement variants modified both the final verdict and trace scores. The official results showed that ranking-only changes were insufficient to achieve the best English or Spanish results, and full-replacement strategies were risky, especially for Spanish.

4.3. Verdict Aggregation

The final verdict was derived from a combination of claim-level prediction, trace verdict distribution, and confidence signals. We treated verdict aggregation as a decision layer operating above the trace scores. This design allowed the system to rank traces and select the final label based on related, but not identical, signals.

For supervised variants, the decision layer used class probabilities from Logistic Regression and applied thresholded rules to avoid overpredicting the majority class. The thresholding strategy was useful because numerical claim verification frequently involves uncertainty and evidence insufficiency. In such cases, forcing a binary `True`/`False` decision can be harmful, while `Conflicting` may be the more appropriate label.

For LLM-assisted variants, the decision layer used Mistral as a verifier-style judge over selected uncertain cases. The LLM returned a candidate final verdict, trace scores, and a confidence value. In the final English configuration, the predicted verdict was accepted only when the confidence condition was met. This made the LLM a confidence-aware post-processing component rather than an unrestricted replacement for the supervised or rule-based prediction.

4.4. Supervised Thresholded Verifier

The supervised thresholded family used TF-IDF representations and Logistic Regression. The claim-level input included the claim, available evidence, selected reasoning-trace information, and summary features from the trace verdict distribution. Logistic Regression produced probabilities for the official labels, and thresholded rules selected the final verdict based on class probability and margin conditions.

The supervised thresholded verifier, selected as the best Spanish system, belonged to this family. It was trained using training and validation data and applied conservative thresholding to the final decision. This configuration achieved the best confirmed Spanish score. We describe it as a supervised classical verifier because it was trained on official labeled data using standard text features and calibrated decision rules. It should not be described as an LLM-based or neural verifier.

This family was particularly suitable for Spanish because it preserved a more stable relationship between observed label distribution, model probabilities, and decision thresholds. In contrast, LLM-assisted post-processing changed too many Spanish decisions, thereby reducing the official score. The Spanish result, therefore, supports the use of language-specific calibration rather than a single shared verification strategy.

4.5. Confidence-Aware LLM-Assisted Verifier

For English, the best configuration was the confidence-aware LLM-assisted verifier, which used Mistral as a non-fine-tuned verifier over selected uncertain cases. The model was not trained or fine-tuned for the task. Instead, it received the claim, evidence, candidate reasoning traces, and candidate verdicts, and returned a JSON object containing a proposed final verdict, trace scores, a confidence value, and a brief explanation.

The LLM output was not accepted unconditionally. The final English configuration changed the final verdict only when the confidence criterion was satisfied, while also using valid LLM-generated trace scores when their length matched the number of reasoning traces. This design reduced the risk

of uncontrolled over-correction. The approach was motivated by the observation that some English examples contained uncertain or conflicting trace signals where a verifier-style reassessment could help identify whether the claim was better represented as `True`, `False`, or `Conflicting`.

The expanded English LLM-assisted verifier increased the number of Mistral-reviewed examples and used a higher confidence threshold. It did not improve the official score. This result suggests that the benefit of LLM-assisted post-processing was concentrated in a smaller subset of uncertain English examples and that expanding the review set may introduce additional noise even under confidence control.

4.6. Language-Specific Configuration

The English and Spanish final systems were selected independently. This decision was based on official validation feedback rather than on a predefined assumption that one architecture would transfer across languages. English obtained its best confirmed score with confidence-aware LLM-assisted verification. Spanish obtained its best confirmed score with supervised training and validation thresholding.

The difference between the two languages is an important methodological outcome. The same LLM-assisted family that helped English marginally did not improve Spanish. In Spanish, full-replacement and confidence-aware LLM variants substantially reduced the score. This suggests that multilingual numerical claim verification requires language-specific calibration, especially when an external LLM is used without fine-tuning.

5. Experimental Setup

5.1. Official Datasets and Languages

We used the official Task 2 data released by the organizers for English and Spanish. Arabic was part of the official task, but it was not included in the systems we submitted. Each training and validation example includes the claim, the associated evidence, 20 LLM-generated reasoning paths, the candidate verdict for each reasoning path, and the corresponding gold label, when available. The final test files preserve the same reasoning-trace structure, requiring the system to output both a ranking of reasoning traces and a final verdict.

Our participation focused on two languages with different behaviors during validation and final submission. English was more responsive to confidence-aware LLM-assisted correction over selected uncertain cases. Spanish achieved better results with supervised training and validation thresholding. Therefore, the final submissions were selected independently by language instead of enforcing a single multilingual configuration.

5.2. Development and Final Evaluation Protocol

The development process was organized around official-format validation and official evaluation feedback. Before uploading each file, we checked that the number of predictions matched the expected number of examples, that each final verdict belonged to the official label set, and that candidate verdicts, reasoning traces, and trace-ranking scores were aligned for every example. This validation step was important because an otherwise reasonable model output could be rejected or incorrectly scored if the JSON structure was inconsistent.

For development analysis, we used local scores and validation outputs when available. However, the final results reported in this working note correspond only to the official scores shown for accepted submissions. Since the leaderboard did not provide the full per-metric breakdown for all submitted variants, we report the official score and avoid inferring macro-F1, Recall@5, MRR@5, or classwise F1 from the official aggregate score.

5.3. Reproducibility Details

The supervised variants were implemented with scikit-learn using TF-IDF features and Logistic Regression. The Spanish supervised thresholded verifier used word n-gram TF-IDF features with up to 80,000 features, n-grams from 1 to 2, a minimum document frequency of 2, and a Logistic Regression classifier with balanced class weights, $C = 2.0$, and a maximum of 2,000 iterations. The final Spanish decision layer used thresholded class probabilities and margin conditions to select the final verdict.

The LLM-assisted variants used the Mistral API with the `mistral-large-latest` model. The model was accessed through the chat-completions endpoint and was not fine-tuned. The verifier prompt included the claim, available evidence snippets, candidate reasoning traces, and candidate verdicts. The model was instructed to return only valid JSON containing a final verdict, one trace score per reasoning trace, a confidence value, and a brief reason. The calls used deterministic decoding with temperature 0.0. The robust runner used a maximum of 900 output tokens, a 120-second timeout, and retry handling for rate-limit and server errors.

For the final English LLM-assisted submission, 100 uncertain cases were selected for Mistral review, and all 100 returned valid responses. Verdict changes were accepted only when the confidence value was at least 0.65; valid Mistral trace scores were used when their length matched the number of reasoning traces. The expanded English LLM-assisted variant reviewed 300 uncertain cases, of which 297 produced valid responses, and used a higher confidence threshold of 0.90. The Spanish LLM-assisted variants reviewed 100 selected cases and obtained 100 valid responses, but they were not selected as final systems because they reduced the official score.

All notebooks were executed in a Google Colab environment with Google Drive storage. No local LLM inference, GPU-based fine-tuning, or task-specific neural training was performed. The Mistral component relied on API calls, and the exact monetary cost of these calls was not logged; therefore, we report the number of reviewed cases instead of an API cost estimate.

5.4. Code Availability and Baseline Comparison

The implementation was developed in Google Colab notebooks with Google Drive storage. At the time of the camera-ready revision, the code was not available in a public GitHub repository. To improve reproducibility, we added the main implementation details to the manuscript, including the supervised model configuration, the Mistral model name, API access, decoding setting, number of reviewed cases, confidence thresholds, and execution environment.

We compared our English development configuration against the official English development baseline released by the organizers. The official English baseline obtained a local development proxy score of 0.425, computed as the average of Recall@5 and macro-F1, while our English supervised thresholded configuration obtained 0.436. This improvement was mainly due to a higher macro-F1, although the official baseline had a slightly higher Recall@5. The Spanish development data did not have a directly comparable official baseline file in the available materials: the downloaded official baseline contained 1,600 English examples and had no overlap with the Spanish validation set. Therefore, Spanish comparisons are reported only against internal development variants and not as an official baseline comparison.

Table 2
Development baseline comparison. Scores are local development estimates and not final test leaderboard scores.

Language	Method	Recall@5	Macro-F1	Proxy score
English	Official development baseline	0.312	0.537	0.425
English	Supervised thresholded verifier	0.302	0.570	0.436
Spanish	Majority baseline	0.177	0.353	0.265
Spanish	Numeric heuristic baseline	0.168	0.351	0.260
Spanish	Supervised thresholded verifier	0.267	0.583	0.425

5.5. Variant Families Evaluated

The submitted configurations were grouped into methodological families rather than treated as a chronological sequence of notebook versions. Table 3 summarizes the main families.

Table 3
Variant families evaluated for Task 2.

Family	Description	English	Spanish
Classical thresholding	TF-IDF and Logistic Regression with calibrated verdict thresholds.	Yes	Final
Claim-probability ranking	Trace scores based on predicted class probabilities.	Yes	Yes
Soft-conflicting calibration	Increased sensitivity toward `Conflicting`.	Exploratory	Yes
Confidence-aware LLM	Mistral changes are accepted only under confidence constraints.	Final	Non-final
LLM ranking-only	Replaced trace-ranking scores; preserved final verdict.	Non-final	Non-final
LLM full-replacement	Replaced final verdict and trace scores.	Non-final	Non-final
Expanded LLM review	Expanded Mistral review for uncertain English cases.	Non-final	No

The classical thresholding family was based on supervised claim-level prediction. It used text features and class probabilities to select the final label under conservative threshold and margin conditions. The confidence-aware LLM-assisted family used Mistral as an external non-fine-tuned verifier over selected uncertain examples. The ranking-only and full-replacement variants were designed to test whether performance gains came mainly from trace reordering or from final verdict correction.

5.6. Final Candidate Selection

The final candidate for each language was selected using the official score. For English, the best confirmed configuration was a confidence-aware LLM-assisted verifier. This variant used Mistral for selected uncertain cases and accepted final-verdict changes only under confidence control. For Spanish, the best confirmed configuration was a supervised thresholded verifier. This variant used supervised training and validation thresholding and did not rely on Mistral for the final decision.

6. Results

6.1. Official English Result

The final English submission was the confidence-aware LLM-assisted verifier. It achieved an official score of 0.395 and was applied to selected uncertain cases. It improved over the strongest classical English configuration, but the improvement remained marginal.

The expanded English LLM-assisted verifier, which reviewed more uncertain cases with Mistral and used a higher confidence threshold, obtained 0.395 and did not improve over the confidence-aware LLM-assisted verifier. This indicates that increasing the number of LLM-reviewed examples did not necessarily improve the final score. The useful LLM-assisted corrections appeared to be concentrated in a smaller subset of uncertain cases.

6.2. Official Spanish Result

The final Spanish submission was a supervised thresholded verifier. It achieved an official score of 0.458. This configuration used supervised training and validation thresholding based on TF-IDF and Logistic Regression.

Unlike English, the best Spanish result did not come from an LLM-assisted variant. The Spanish Mistral-based configurations did not improve over the supervised thresholded model. In particular, the

Spanish confidence-aware LLM-assisted variant and the Spanish LLM full-replacement variant both obtained 0.420, while the Spanish LLM ranking-only variant obtained 0.451. These results suggest that the LLM-assisted verifier over-corrected or miscalibrated Spanish examples when applied without language-specific fine-tuning.

Table 4
Official confirmed results.

Language	Final method	Score
English	confidence-aware LLM-assisted verifier	0.395
Spanish	supervised thresholded verifier	0.458

6.3. Variant Analysis

Table 5 summarizes the main final and non-final submissions; entries marked as "Analysis" were not selected as final systems. The results show three relevant patterns. First, English benefited slightly from confidence-aware LLM-assisted verification, but expanding LLM review to more examples did not improve the score. Second, Spanish benefited more from supervised thresholding than from LLM-assisted correction. Third, ranking-only variants were insufficient to achieve the best final results, suggesting that trace ranking and final verdict selection must be optimized together.

Table 5
Main final and non-final submissions.

Lang.	Method	Score	Decision
EN	Confidence-aware LLM-assisted verifier	0.395	Final
EN	Expanded LLM-assisted verifier	0.395	Analysis
EN	LLM full-replacement variant	0.395	Analysis
EN	LLM ranking-only variant	0.393	Analysis
EN	Validation-like classical variant	0.393	Analysis
ES	Supervised thresholded verifier	0.458	Final
ES	Claim-probability supervised variant	0.456	Analysis
ES	Spanish LLM ranking-only variant	0.451	Analysis
ES	Spanish confidence-aware LLM-assisted variant	0.420	Analysis
ES	Spanish LLM full-replacement variant	0.420	Analysis
ES	Soft-conflicting calibration variant	0.392	Analysis

6.4. Language-Level Comparison

The official results indicate that English and Spanish required different final configurations. English obtained its best score from the confidence-aware LLM-assisted verifier, whereas Spanish obtained its best score from the supervised thresholded verifier. This contrast suggests that the effectiveness of verifier-style post-processing depends on the language, the reliability of the LLM's judgment, and the calibration of the final verdict-selection layer.

The Spanish results are particularly informative because the LLM-assisted variants were not merely neutral; some of them substantially reduced the score. This suggests that changing the final verdict based on an external LLM judge can be harmful when the judge is not calibrated to the target language or the distribution of numerical claim labels. By contrast, the supervised Spanish model likely captured language-specific regularities from the official training and validation data.

The English results show a different pattern. The LLM-assisted verifier provided a small improvement over previous English configurations, but additional LLM processing in the expanded LLM-assisted verifier did not improve performance. This suggests that confidence-aware LLM assistance can be useful for carefully selected examples but should not be expanded indiscriminately.

7. Discussion

The official results show that Task 2 should not be treated as a standard text classification problem. The presence of multiple reasoning traces changes the decision process: the system must determine not only which label is most likely but also which reasoning paths are sufficiently trustworthy to support that label. Our experiments indicate that trace ranking and verdict selection are related but distinct components. Modifying only the trace-ranking scores did not consistently produce the best results, while replacing both the verdict and trace scores without sufficient calibration was risky.

The English final configuration benefited from confidence-aware LLM-assisted verification. The key factor was not the use of an LLM by itself, but the restricted way in which the LLM output was incorporated. In the confidence-aware LLM-assisted verifier, Mistral was used as a verifier over selected uncertain cases, and its proposed verdict was accepted only under confidence control. This design limited the risk of systematic over-correction. The result improved over previous English configurations, but the margin remained small. Therefore, the English result should be interpreted as evidence that confidence-aware LLM assistance can help in selected cases, not as evidence that broad LLM replacement is sufficient.

The expanded English LLM-assisted verifier is informative because it reviewed more uncertain cases with Mistral and used a higher confidence threshold. Still, it did not improve the official score. This suggests that the useful LLM corrections were concentrated in a limited subset of uncertain examples. Once the review set was expanded, additional LLM decisions may have introduced noise or changed examples that were not beneficial for the final metric. In numerical verification, more LLM intervention is not necessarily better; the selection of cases where the LLM is allowed to intervene is part of the verifier design.

The Spanish results show a different pattern. The best Spanish configuration was not LLM-assisted. Instead, the supervised thresholded verifier achieved the highest confirmed score using supervised training and validation thresholding. This suggests that for Spanish, the supervised model captured useful regularities from the official data and maintained a more stable decision boundary than the external LLM judge did. The Mistral-based Spanish variants, particularly the Spanish confidence-aware LLM-assisted variant and Spanish LLM full-replacement variant, substantially reduced the score. This behavior is consistent with over-correction: the LLM judge changed decisions that were better handled by the supervised thresholded verifier.

The contrast between English and Spanish supports the use of language-specific configurations. Although both languages share the same task format, their best-performing strategies differed. English achieved its best result with a confidence-aware LLM-assisted verifier, whereas Spanish achieved its best result with supervised thresholding. This does not necessarily mean that LLMs are ineffective for verifying numerical claims in Spanish. Rather, it indicates that an external non-fine-tuned LLM judge should not be used as a direct replacement without calibration to the language, evidence style, and label distribution.

Another relevant finding is that the `Conflicting` label requires careful treatment. In numerical claim verification, `Conflicting` may represent evidence insufficiency, ambiguity, disagreement across sources, or traces that do not resolve the decisive numerical relation. Increasing sensitivity to `Conflicting` can help when evidence is genuinely inconclusive, but it can also hurt if the model assigns `Conflicting` to cases where the evidence actually supports or contradicts the claim. The Spanish soft-conflicting calibration variant illustrates this risk because it reduced the official score.

8. Error Analysis

Detailed official per-metric breakdowns were not available for all submitted variants. Therefore, this error analysis focuses on the observed behavior of our submitted systems and development variants rather than on instance-level official diagnostics. The main failure patterns were associated with four components of our pipeline: numerical and temporal mismatch handling, selection of weak but

plausible reasoning traces, overuse or underuse of the `Conflicting` label, and over-correction by the LLM-assisted verifier, especially in Spanish.

8.1. Numerical Mismatch

A frequent source of error in numerical claim verification is the mismatch between the number expressed in the claim and the number supported by the evidence. A trace may mention the correct entity, event, or topic while failing to resolve the exact quantity. This is especially problematic when claims involve percentages, budgets, rates, differences, or accumulated totals. A reasoning trace that is semantically related to the claim may still be invalid if it uses a different denominator, a different unit, or a different aggregation level.

This type of error affects both trace ranking and verdict selection. If the trace addresses the correct topic but does not meet the numerical condition, assigning it a high score may lead the system to an incorrect verdict. Future systems should therefore incorporate explicit numerical comparison features, including unit normalization, percentage matching, year matching, and consistency checks between claim numbers and evidence numbers.

8.2. Temporal Reasoning Errors

Temporal expressions are another source of instability. Many numerical claims depend on a specific year, month, fiscal period, election cycle, or policy implementation date. A trace can be misleading if it uses evidence from a different period or mixes historical and current values. In such cases, the final verdict may change depending on whether the claim is interpreted as current, historical, cumulative, or comparative.

The LLM-assisted variants sometimes improved cases where the trace evidence clearly contradicted a temporal claim. However, they also risked over-correction when the evidence was ambiguous or when several traces spanned different time windows. Temporal reasoning remains a key limitation because it requires aligning the claim, evidence, and reasoning trace at the level of dates and periods, not merely at the level of topic similarity.

8.3. Evidence Insufficiency and Ambiguity

The `Conflicting` label is difficult to apply because it may arise from several situations: insufficient evidence, contradictory evidence, incomplete reasoning traces, or claims whose exact wording cannot be verified in the available snippets. In these cases, the safest verdict may be `Conflicting`, but overusing this label can reduce performance if the evidence actually supports a more specific `True` or `False` decision.

The Spanish experiments clearly showed this risk. Variants that increased sensitivity toward `Conflicting` did not outperform the supervised thresholded model. This suggests that `Conflicting` requires calibrated decision boundaries rather than simple expansion. A useful verifier must distinguish between genuine evidential ambiguity and cases where a trace fails, but another trace provides sufficient evidence for a non-conflicting verdict.

8.4. Incorrect Trace Selection

Trace selection errors occur when a plausible but weak reasoning trace receives a high score. Such traces may contain fluent explanations and relevant entities, but they may not address the decisive numerical relation. Conversely, a shorter or less fluent trace may contain the decisive contradiction or support. This makes trace ranking more complex than ranking by surface plausibility.

The ranking-only variants did not produce the best final results, indicating that improving trace scores without changing the final verdict was insufficient. This suggests that trace ranking and verdict aggregation should be jointly optimized. Future work should investigate models that score a trace according to both its evidential relevance and its consistency with the final verdict.

8.5. Verdict Aggregation Conflicts

A single claim may contain reasoning traces with different candidate verdicts. Majority-based aggregation is not always reliable because several traces can repeat the same mistaken interpretation. However, accepting a minority trace can also be risky if it is unsupported or overconfident. The final decision layer must therefore balance majority agreement, class probabilities, trace quality, and confidence.

The English confidence-aware LLM-assisted verifier configuration partially addressed this problem by allowing LLM-assisted changes only under confidence control. In Spanish, however, LLM-assisted verdict changes reduced performance, suggesting that the aggregation strategy must be language-specific and calibrated against development behavior.

8.6. Over-Correction by LLM-Assisted Variants

The strongest negative effect observed in our experiments was over-correction by the Spanish LLM-assisted variants. The Spanish supervised thresholded verifier achieved an official score of 0.458, whereas the Spanish confidence-aware and full-replacement LLM-assisted variants decreased to 0.420. This drop suggests that the LLM judge changed verdicts that were better handled by the supervised thresholded decision layer. In other words, the failure was not caused by invalid output formatting, but by miscalibrated verdict changes: the LLM produced plausible judgments that were not sufficiently aligned with the Spanish label distribution, the task-specific evidence format, or the official evaluation objective.

For this reason, LLM-assisted verification should be treated as a controlled component rather than as a full replacement for supervised or calibrated decision rules. Confidence thresholds, case selection, and language-specific validation are essential. The English results show limited benefit under such constraints, while the Spanish results indicate that these constraints were insufficient to guarantee transfer.

9. Limitations

This participation has several limitations. First, the LLM-assisted verifier relied on a single external model, `mistral-large-latest`, accessed through the Mistral API and used without fine-tuning. We did not evaluate alternative LLMs or task-specific neural verifiers. Therefore, the conclusions about LLM-assisted verification are limited to this specific non-fine-tuned verifier setting.

Second, the final results are reported using the official scores because complete per-metric breakdowns were not available for all submitted variants. We therefore avoid inferring macro-F1, Recall@5, MRR@5, or classwise F1 from the official aggregate score. Baseline comparisons are limited to development estimates: English was compared against the official English development baseline, whereas Spanish was compared only against internal development baselines because a directly comparable official Spanish baseline file was not available in the materials we used. This limits the granularity of the quantitative analysis.

Third, the implementation was developed in Google Colab notebooks and was not available in a public GitHub repository at the time of the camera-ready revision. To mitigate this limitation, we added the main reproducibility details to the manuscript, including the model configuration, Mistral API access, decoding settings, confidence thresholds, the number of reviewed cases, and the execution environment.

Fourth, the Arabic track was not addressed. The official task included English, Spanish, and Arabic, but our submitted systems focused only on English and Spanish. Therefore, the conclusions about language-specific behavior should not be generalized to Arabic.

Fifth, the improvement in English from LLM-assisted verification was marginal. Although the confidence-aware LLM-assisted verifier produced the best English score, the expanded LLM-assisted verifier did not improve it. This suggests that the benefit of the LLM-assisted verifier was limited and sensitive to case selection.

Sixth, Spanish did not benefit from the LLM-assisted variants. The best Spanish result came from supervised thresholding, while Mistral-based confidence-aware and full-replacement variants reduced performance. This indicates that multilingual LLM-assisted verification requires stronger language-specific calibration or fine-tuning.

Finally, the approach depends on the quality of the reasoning traces provided by the organizers. If the available traces omit the decisive numerical comparison or repeat the same mistaken interpretation, both ranking and verdict aggregation become more difficult. Future systems should consider evidence-aware trace generation, explicit numerical consistency checking, and verifier models trained on trace-level supervision.

10. Conclusions and Future Work

This working note presented the SINAI-UGPLN participation in CheckThat! 2026 Task 2 for English and Spanish. We modeled the task as a joint reasoning-trace ranking and verdict-selection problem and compared supervised thresholded configurations with confidence-aware LLM-assisted verification. The final English submission was the confidence-aware LLM-assisted verifier, which used a non-fine-tuned LLM under confidence control and achieved an official score of 0.395. The final Spanish submission was the supervised thresholded verifier, which used supervised training and validation thresholding and achieved an official score of 0.458.

The main finding is that language-specific behavior matters. English benefited slightly from confidence-aware LLM-assisted verification, but expanding the LLM review set did not improve the score. Spanish performed best with supervised thresholding, and LLM-assisted variants reduced performance. These results indicate that verifier strategies for numerical claim verification should be calibrated separately by language and should not assume that an external LLM judge will transfer effectively across languages.

Future work should focus on trained verifier models that jointly optimize trace ranking and final verdict selection. Promising directions include evidence-aware trace reranking, explicit numerical and temporal consistency features, multilingual calibration, fine-tuning verifier models with trace-level supervision, and controlled comparisons with alternative LLMs under the same confidence-aware verification protocol. Another relevant direction is to distinguish evidential insufficiency from genuine contradiction, especially for the `Conflicting` label. These improvements would help build more reliable numerical fact-checking systems that can use LLM-generated reasoning traces without over-relying on them.

Acknowledgments

This work is funded by the Ministerio para la Transformación Digital y de la Función Pública and Plan de Recuperación, Transformación y Resiliencia – Funded by EU – NextGenerationEU within the framework of the project Desarrollo Modelos ALIA. This work has also been partially supported by Project CONSENSO (PID2021-122263OB-C21), Project MODERATES (TED2021-130145B-I00) and Project SocialTox (PDC2022-133146-C21) funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. Moreover, this research is part of the proposal presented at the Call for Research Project Proposals of the Internal Competitive Fund (FCI) 2023, which was approved on September 14, 2023 (Resolution No. R-CSU-UG-SE34-313-14-09-2023) by the Consejo Superior Universitario of the Universidad de Guayaquil.

The authors declare that they have contributed equally and share authorship roles for this publication.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (OpenAI) and Grammarly to support grammar checking, spelling revision, and language refinement. Mistral was used only as a non-fine-

tuned verifier component in selected experimental variants, as described in the methodology. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content. All official scores, final variant decisions, and methodological claims were verified by the authors against CodaBench outputs and local validation logs. No unverified leaderboard positions, datasets, or metric values were generated or reported.

References

[1] V. V, V. Setty, P. Chungkham, A. Anand, F. Alam, Overview of the CLEF-2026 CheckThat! lab task 2 on fact-checking numerical claims, CLEF 2026, Jena, Germany, 2026.

[2] P. Chungkham, V. V, V. Setty, A. Anand, Think Right, Not More: Test-Time Scaling for Numerical Claim Verification, in: Findings of the Association for Computational Linguistics: EMNLP 2025, 2025, pp. 24345–24363.

[3] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-Consistency Improves Chain of Thought Reasoning in Language Models, in: Proceedings of the International Conference on Learning Representations, 2023.

[4] CheckThat! Lab, CLEF2026 CheckThat Lab Task 2 Code, Scorer, Format Checker, and Baseline Scripts, https://gitlab.com/checkthat_lab/clef2026-checkthat-lab/-/tree/main/task2/Code, 2026. Accessed: 2026-05-27.

[5] V. V, A. Anand, A. Anand, V. Setty, QuanTemp: A Real-World Open-Domain Benchmark for Fact-Checking Numerical Claims, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 650–660.

[6] Mistral AI, Mistral AI Documentation, https://docs.mistral.ai/, 2026. Accessed: 2026-05-27.

[7] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training Language Models to Follow Instructions with Human Feedback, Advances in Neural Information Processing Systems 35 (2022) 27730–27744.