AKSH at CheckThat! 2026: A Tri-Model Ensemble Approach for Fact-Checking Numerical Claims

Notebook for the CheckThat! Lab at CLEF 2026, Task 2

Syed Hamza Raza[*,†], Abdul Karim[†], Faisal Alvi and Abdul Samad

Department of Computer Science, Dhanani School of Science and Engineering, Habib University, Karachi, Pakistan

Abstract

This paper describes the system submitted by team AKSH to the CLEF 2026 CheckThat! Lab for Task 2 (Fact-Checking Numerical Claims). We designed a heterogeneous, tri-model ensemble composed of three independently trained verifier models: a fine-tuned XLM-RoBERTa sequence classifier (Approach 1), a LLaMA-3.2-1B-Instruct generative reward model adapted via 4-bit QLoRA (Approach 2), and a zero-shot GPT-4o-mini scoring pipeline whose raw logits are normalised via softmax (Approach 3). Evidence for each reasoning trace is retrieved with BM25Okapi (Top-1 snippet), and the three probability distributions are fused using language-specific weights determined by grid search on the official development set. On the hidden test set, our ensemble achieved a Macro Avg F1 of **0.5817** for English (rank 15/22), **0.3031** for Spanish (rank 16/16), and **0.3557** for Arabic (rank 15/15). Our system demonstrated high precision in identifying negative claims, achieving a **False F1 score of 0.8915 (rank 1/22)** for the English track. We provide a detailed analysis of our ensemble's performance and the limitations of our tri-model approach.

Keywords

fact-checking, numerical claims, reasoning trace ranking, ensemble, XLM-R, LLaMA, GPT-4o-mini, CLEF CheckThat!

1. Introduction

The proliferation of quantitative misinformation — false or misleading claims involving specific numbers, statistics, and temporal expressions — poses an acute challenge for automated fact-checking systems. Such claims require not only lexical matching against evidence but also numerical reasoning, unit normalisation, and the ability to detect subtle logical fallacies embedded within otherwise plausible prose. The CLEF 2026 CheckThat! Lab [1, 2] provides a rigorous multilingual benchmark for automating key steps of the fact-checking pipeline, continuing a series of shared evaluations dating back to 2018 [3, 4].

Task 2 of this year's edition [5] frames numerical claim verification as a *test-time scaling* problem. Recent advances in large language models demonstrate that allocating additional compute during inference—specifically through sampling-based search strategies—can significantly enhance reasoning capabilities without altering the base model weights [6, 7]. *Best-of-N* (BoN) sampling is a common paradigm in this space, where a model generates **N** candidate reasoning paths, and a dedicated verifier selects the most promising trajectory [8]. In this task, given a claim, retrieved evidence snippets, and a pool of diverse reasoning traces generated by an LLM at varying temperatures, participating systems must act as this verifier, ranking the traces by their utility to output a final Best-of-N decision.

Our Task 2 system, submitted under the team name **AKSH**, targets the core difficulty of the task: no single model architecture reliably dominates across three typologically and morphologically distinct languages (English, Spanish, and Arabic) when the label distribution is heavily skewed toward the *False* class. Encoder-only transformers such as XLM-RoBERTa excel at bidirectional contextual pattern recognition [9] and benefit from supervised fine-tuning for numerical claim verification [10]. Building on this finding, our Approach 1 fine-tunes the multilingual XLM-RoBERTa model specifically on labelled

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

[*]Corresponding author.

[†]These authors contributed equally.

sr09016@st.habib.edu.pk\hamzaraza2004@gmail.com (S. H. Raza); ak08731@st.habib.edu.pk (A. Karim); faisal.alvi@sse.habib.edu.pk (F. Alvi); abdul.samad@sse.habib.edu.pk (A. Samad)

reasoning trace data. Decoder-only generative models such as LLaMA-3.2-1B-Instruct, fine-tuned with low-rank adaptation, can reason sequentially over long context and assign implicit reward signals to candidate traces. Proprietary models such as GPT-4o-mini provide strong zero-shot logical reasoning without requiring any local training budget. Rather than selecting one paradigm, we combine all three via a soft-voting ensemble whose language-specific mixture weights are optimised by grid search on the official development set, allowing each component to contribute where it is strongest.

A secondary challenge we encountered during submission was an engineering one: a JSON parsing error in our inference pipeline caused the textual content of reasoning traces to be lost for the Spanish and Arabic outputs. We detail this bug, the post-processing workaround we deployed, and its precise impact on the official evaluation metrics in Section 5 and Section 6.

The remainder of this paper is structured as follows. Section 2 formally defines Task 2. Section 3 situates our work within prior CheckThat! research. Section 4 describes our full methodology, from preprocessing through ensemble optimisation. Section 5 presents results on both the development and hidden test sets. Section 6 provides an error analysis of our Spanish and Arabic submissions. Section 7 concludes with directions for future work.

2. Task Description

Task 2 of the CLEF 2026 CheckThat! Lab is titled *Fact-Checking Numerical and Temporal Claims* [5]. It frames numerical claim verification as a *test-time scaling* problem, building on the 2025 numerical-claim task while introducing an explicit reasoning-trace ranking objective that evaluates not only factual accuracy but also the quality of the model's underlying justification.

2.1. Problem Formulation

For each instance, a system receives the following inputs:

- **Claim**: a naturally occurring statement containing numerical quantities and/or temporal expressions (e.g., *"Inflation rose by 8% in 2022"*).
- **Evidence**: a list of verified factual snippets retrieved from the web that are relevant to the claim.
- **Reasoning Traces**: a pool of candidate justifications generated by an LLM at varying temperature values to induce diversity, followed by a de-duplication step to remove redundant traces. Each trace contains a natural-language justification and a predicted verdict.
- **Verdict List**: the per-trace final verdicts predicted by the generating LLM.
- **Verdict**: A single aggregated verdict provided by the dataset organizers representing the baseline prediction of the generating LLM.

During training and validation, the models are optimized against the following ground-truth target:

- **Label**: The veracity category (*True*, *False*, or *Conflicting*) assigned to the claim by human annotators.

Systems must produce two outputs per instance:

1. **Ranked Reasoning Traces**: the full pool of traces re-ordered by their estimated utility in reaching the correct verdict (Best-of-N ranking).
2. **Verdict_BoN**: a single final verdict label derived from the top-ranked trace(s), drawn from three classes: *True*, *False*, or *Conflicting*.

2.2. Languages and Data

The task covers three languages: English (EN), Spanish (ES), and Arabic (AR). Table 1 reports the per-class label counts for all three splits (train, development, and hidden test) as computed directly from the released JSON files. Several distribution properties are critical to understanding our system's behaviour.

English maintains a reasonably consistent class distribution across splits, with *False* comprising 57–70% of instances and *True* and *Conflicting* each accounting for roughly 10–24%. The hidden test set is noticeably more skewed toward *False* (70.3%) than the development set (57.1%), which contributes to the minority-class recall challenges discussed in Section 6.1.

Spanish shows the most severe imbalance across all three languages: *False* accounts for 84% of both train and development instances. Although the hidden test set is more balanced (70.5% *False*, 21.8% *True*), our model was trained and tuned on the heavily skewed distribution, which directly causes the *True* F1 collapse to 0.0000 on the test set (Section 6.1).

Arabic contains no *Conflicting* instances whatsoever in either the training or development sets, meaning no model in our pipeline ever learned to predict this class for Arabic. The hidden test set introduces 32 *Conflicting* instances (6.3%), for which our system produces zero correct predictions by construction.

Table 1
Per-class label distribution across train, development (dev), and hidden test splits for all three languages. Counts and percentages are computed from the released JSON gold-label files. † No *Conflicting* instances exist in the Arabic train or dev splits; the 32 test instances of this class were unseen by all models.

Lang.	Split	False	True	Conflicting	Total
EN	Train	3,717 (58.1%)	1,175 (18.4%)	1,508 (23.6%)	6,400
	Dev	913 (57.1%)	304 (19.0%)	383 (23.9%)	1,600
	Test	1,799 (70.3%)	488 (19.1%)	271 (10.6%)	2,558
ES	Train	1,888 (84.1%)	152 (6.8%)	206 (9.2%)	2,246
	Dev	473 (84.2%)	38 (6.8%)	51 (9.1%)	562
	Test	821 (70.5%)	254 (21.8%)	89 (7.6%)	1,164
AR	Train	1,444 (55.4%)	1,164 (44.6%)	0† (0.0%)	2,608
	Dev	361 (55.4%)	291 (44.6%)	0† (0.0%)	652
	Test	218 (42.7%)	261 (51.1%)	32 (6.3%)	511

2.3. Evaluation Metrics

The official evaluation uses two families of metrics.

Classification metrics (primary): Macro-averaged F1 (`Macro Avg F1`) computed over the three verdict classes on the `Verdict_BoN` output. Per-class F1 scores (`True F1`, `False F1`, `Conflicting F1`) are reported as supporting diagnostics.

Information-retrieval metrics (supporting): `Precision@`k and `Recall@`k measure whether the highest-quality reasoning traces are ranked near the top of the predicted list. `Average Score` combines classification and retrieval signals into a single composite figure. These metrics depend on the textual content of the ranked trace array and are therefore only meaningful when the full trace strings are present in the submission file (see Section 6 for the impact this had on our Spanish and Arabic submissions).

3. Related Work

Our system draws on three strands of prior work: multilingual claim verification, hybrid retrieval for evidence grounding, and the use of LLM-based prompting pipelines as verifier components.

3.1. Numerical Claim Verification at CheckThat!

The 2025 edition of CheckThat! introduced numerical claim fact-checking as Task 3, attracting thirteen participating teams [11]. The dominant pipeline that year decomposed claims using GPT-4o-mini, retrieved evidence with BM25, re-ranked candidates with a cross-encoder, and classified with a fine-tuned NLI model — a multi-stage approach that informed our own evidence retrieval strategy. ClaimIQ [10] directly compared zero-shot LLM prompting against supervised fine-tuning (RoBERTa and LoRA-tuned LLaMA) for the same task, finding that fine-tuning improved validation performance, though the gap narrowed on the hidden test set. This finding motivated our decision to combine fine-tuned and zero-shot components rather than committing to either paradigm alone.

3.2. Encoder vs. Decoder for Claim Classification

A persistent finding across CheckThat! editions is that fine-tuned encoder-only models remain competitive with, and often outperform, larger prompted models when in-domain labelled data is available. At CheckThat! 2023, the OpenFact team showed that while GPT-4 was competitive in zero-shot scenarios, it was surpassed by fine-tuned BERT-family models on in-domain data [12]. Leistra and Caselli demonstrated that language-specific fine-tuning of mDeBERTaV3 outperforms generic multilingual tuning for subjectivity detection [13], a lesson we applied when designing separate ensemble weights for each language. At CheckThat! 2024, CLaC-2 showed that a fully zero-shot instruction-tuned LLM could achieve competitive results on check-worthiness [14], confirming the growing viability of prompt-only pipelines as ensemble members.

3.3. Test-Time Scaling and Best-of-N Sampling

Task 2 directly operationalises recent findings in test-time compute, where increasing compute during inference via sampling and verification significantly enhances LLM reasoning capabilities [6, 7]. A prominent framework in this space is Best-of-N (BoN) sampling, which separates generation from verification: a generator model samples a diverse pool of **N** reasoning trajectories, and a separate reward model or verifier evaluates and selects the optimal path [8]. Our tri-model ensemble functions entirely within this verifier role. By combining cross-architecture probability distributions to rank the candidate reasoning traces, our system attempts to optimise the verification step of the BoN pipeline, which remains a critical bottleneck in scaling test-time compute for complex fact-checking tasks.

3.4. Retrieval-Augmented Verification

Retrieval-augmented generation has become a standard component in fact-checking pipelines. The DEFAULT system at CheckThat! 2024 [15] used a differentiable top-k operator to select authoritative evidence before classification, demonstrating that structured, classification-aware evidence selection significantly improves retrieval performance for rumor verification. On the scientific source retrieval task (CheckThat! 2025, Task 4b), the ATOM system [16] benchmarked various retrieval configurations and found that while BM25 serves as a strong sparse baseline, dense retrievers paired with neural listwise re-rankers consistently achieved the best performance. We adopt a lightweight variant of this principle for Task 2: rather than training a dedicated retrieval model, we use BM25Okapi to extract a single Top-1 evidence snippet per reasoning trace and concatenate it with the claim and trace text as a fixed-length context for our classifiers.

3.5. Ensemble Methods for Multilingual NLP

Ensembles over heterogeneous model families have a strong track record in shared tasks. For instance, TurQUaz at CheckThat! 2024 [17] combined supervised fine-tuning with in-context learning—aggregating predictions via weighted averaging and majority voting—to exploit complementary strengths. More broadly, the shift from pure BERT fine-tuning toward LLM-backed pipelines across the

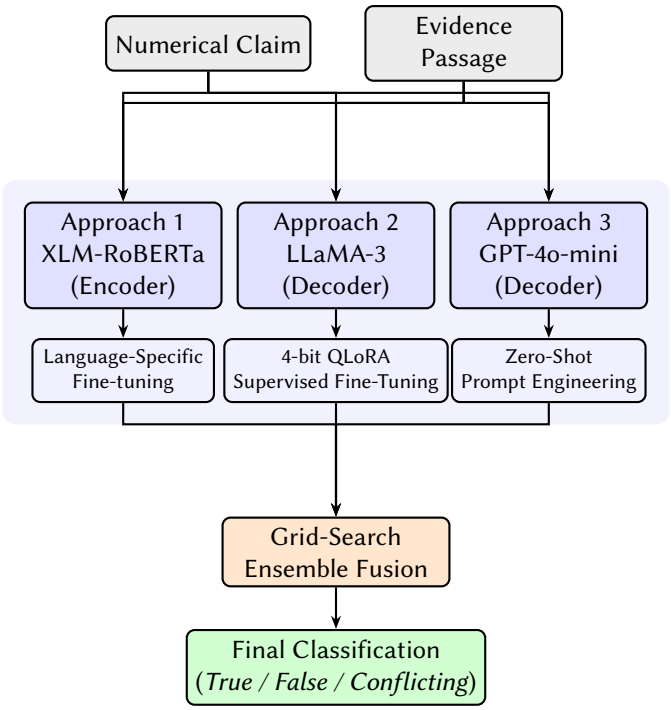

Figure 1: High-level Tri-Model Ensemble Architecture. The claim and evidence passage are independently processed by three parallel approaches; their outputs are combined via grid-search optimised weighted voting to produce the final three-way classification.

last three CheckThat! editions (2023–2025) suggests that neither fine-tuned encoders nor prompted decoders alone represent the ceiling; heterogeneous combinations consistently outperform homogeneous baselines [12, 17, 10]. Our tri-model ensemble — XLM-RoBERTa (encoder), LLaMA-3.2-1B-Instruct (decoder), and GPT-4o-mini (proprietary zero-shot) — is designed to exploit exactly this complementarity, with language-specific mixture weights determined empirically by grid search on the development set.

4. Methodology

Our system is a heterogeneous, tri-model ensemble that combines three independently trained verifier models, each representing a distinct modelling paradigm: an encoder-only sequence classifier (Approach 1), a decoder-only generative reward model (Approach 2), and a proprietary zero-shot scoring pipeline (Approach 3). Figure 1 sketches the full pipeline at a high level. All experiments were conducted on Kaggle GPU notebooks using T4 hardware.

4.1. Preprocessing and Evidence Extraction

All three approaches share the same preprocessing front-end, which performs two operations on every reasoning trace before it is passed to any model.

Trace parsing. Each raw trace string is parsed with a pair of regular expressions that isolate the `Justification` field and the `Label` field (e.g., SUPPORTS, REFUTES, or CONFLICTING). The extracted label is then normalised to one of the three internal classes (`false`, `true`, `conflicting`) by string matching: tokens containing *"refute"* or *"false"* map to `false`; tokens containing *"support"* or *"true"* map to `true`; everything else maps to `conflicting`.

BM25 Top-1 evidence extraction. For each trace, a query is formed by concatenating the claim text with the extracted justification. This query is scored against all evidence snippets in the instance using

`BM25Okapi` from the `rank_bm25` library, with whitespace tokenisation. The single highest-ranked evidence snippet (Top-1) is selected and appended to the model input. This design choice deliberately avoids a dedicated dense retriever in order to keep the pipeline lightweight and reproducible on GPU-constrained hardware.

4.2. Approach 1: XLM-RoBERTa Sequence Classifier

Architecture. Approach 1 fine-tunes `xlm-roberta-base` as a three-class sequence classifier (*False*, *True*, *Conflicting*). The model processes the full context of a reasoning trace bidirectionally, making it well suited to capturing entailment patterns between the claim, the evidence, and the trace justification.

Input format. Each trace is encoded as a single flat string:

```
Claim: {claim} Verdict: {norm_v} Justification: {justification}
Evidence: {top_evidence}
```

This string is tokenised, truncated, and padded to a maximum length of 512 tokens.

Scoring and ranking. The model outputs three logit scores, one per class. These are converted to a probability distribution via softmax. The score assigned to the trace is the softmax probability corresponding to the trace's own normalised verdict label (i.e., the probability the model assigns to the class that the trace claims). Formally, for trace i with normalised verdict $v_i \in \{\textit{false}, \textit{true}, \textit{conflicting}\}$:

$$s_i^{(1)} = \mathrm{softmax}(\mathbf{l}_i)_{v_i} \tag{1}$$

where $\mathbf{l}_i \in \mathbb{R}^3$ is the logit vector for trace i. The trace with the highest score is selected as the best trace, and its normalised verdict becomes the `Verdict_BoN` prediction. We explicitly note that this design intentionally couples the trace's ranking utility with its classification alignment. While traditional Best-of-N paradigms might employ an independent, verdict-neutral reward model to evaluate narrative coherence or logical flow, automated fact-checking necessitates strict factual grounding. Large language models frequently generate highly coherent, persuasive justifications for hallucinated claims. By forcing the trace's ranking score to rely entirely on the verifier's independent agreement with the trace's conclusion, we enforce a strict consistency constraint. This directly prevents articulate but factually incorrect traces from dominating the Best-of-N selection, ensuring that ranking and classification are mutually reinforcing.

Class-weight correction. The training labels are heavily skewed toward *False* (as detailed in Table 1). To prevent the classifier from collapsing to a majority-class baseline, inverse-frequency class weights are applied in the cross-entropy loss during fine-tuning, penalising errors on the minority *True* and *Conflicting* classes more heavily. These weights were calculated based on the label distribution of the respective training set for each language. As discussed in Section 6.1, while this mitigation helps, the extreme imbalance on the hidden test set remains the primary driver of performance degradation for the minority classes.

Training configuration. The classifier is fine-tuned for 4 epochs with a learning rate of 1×10^{-5} (cosine schedule, warmup ratio 0.1), batch size 8 (gradient accumulation 2, effective batch size 16), and weight decay 0.01, using `AdamW` via the Hugging Face `Trainer` API. Class weights are computed as inverse label frequency over the language-specific training split (e.g., for English: 0.577 / 1.801 / 1.404 for *false*/*true*/*conflicting*). Model selection uses the checkpoint with the highest `Macro F1` on the development split, evaluated every epoch. Training uses a fixed seed of 42 for reproducibility across all three languages.

4.3. Approach 2: LLaMA-3.2-1B-Instruct Generative Reward Model

Architecture and quantisation. Approach 2 adapts `meta-llama/Llama-3.2-1B-Instruct` [18] as an implicit generative reward model. Because the full model does not fit in Kaggle GPU memory at full precision, it is loaded in 4-bit NormalFloat quantisation (`nf4`) using `BitsAndBytesConfig`, with double quantisation enabled and compute dtype `float16`. Only a small subset of parameters is updated during training via Quantized Low-Rank Adaptation (QLoRA) [19]; all other weights remain frozen.

QLoRA configuration. LoRA adapters are applied to all attention and MLP projection matrices (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`) with rank $r = 16$, $\alpha = 32$, and dropout 0.05; the base weights remain frozen and only the adapter parameters (bias `"none"`) are trained. Fine-tuning uses a learning rate of 2×10^{-4} with a cosine schedule, effective batch size 8 (per-device batch 2, gradient accumulation 4), gradient checkpointing, and weight decay 0.01, for 2 epochs (12,000 steps) on English and Arabic and 1.5 epochs (10,000 steps, shorter warmup of 500 vs. 600 steps) on Spanish, reflecting that language's smaller training set. Training uses standard next-token prediction loss over the full formatted prompt (i.e., loss is not masked to the response span), with checkpoint selection by `eval_loss` and a fixed seed of 42.

Prompt format. The model receives a structured chat-style prompt in the LLaMA-3 instruction format, using the <|`begin_of_text`|>, <|`start_header_id`|>, and <|`eot_id`|> control tokens:

> `[system]` You are a fact-checking verifier. Given a claim, evidence, and a reasoning trace, output the single correct verdict: True, False, or Conflicting.
> `[user]` Claim: {claim}
> Evidence: {top_evidence}
> Trace Verdict: {trace_verdict}
> Justification: {justification}
> Verdict: `[assistant]`

Next-token probability scoring. Rather than reading the full generated output, the score for each trace is derived from the next-token probability distribution at the final prompt position. The tokeniser produces multiple valid surface forms of each verdict word (e.g., `"True"`, `" True"`, `"\nTrue"`); the token IDs for all variations of the trace's normalised verdict are collected, and their softmax probabilities are summed:

$$s_i^{(2)} = \sum_{t \in \mathrm{ids}(v_i)} \mathrm{softmax}(\mathbf{z}_{-1})_t \tag{2}$$

where $\mathbf{z}_{-1}$ is the logit vector at the last token position and $\mathrm{ids}(v_i)$ is the set of token IDs corresponding to all surface forms of verdict v_i. The trace with the highest score is ranked first, and its verdict becomes the `Verdict_BoN` prediction.

4.4. Approach 3: GPT-4o-mini Zero-Shot Scoring

Setup. Approach 3 calls `openai/gpt-4o-mini` through the OpenRouter API, without any fine-tuning, using the `AsyncOpenAI` client with a concurrency limit of 3 simultaneous requests and a per-request timeout of 45 seconds. All calls use `temperature=0.0` and the `json_object` response format to ensure structured, parseable output. Failed or malformed API calls are retried up to 6 times, with linearly increasing backoff (5s $\times$ attempt) on rate-limit errors and a fixed 2-second delay on other failures; traces that still fail after all retries are marked `ERROR_TIMEOUT` and excluded from the vote.

While Approach 3 is natively zero-shot, its final trace ranking is conditioned on the class-probability priors derived from Approach 1, ensuring that our GPT-4o-mini pipeline remains grounded in the supervised signals learned by our encoder-only model.

Batch evaluation prompt. All reasoning traces for a given claim are evaluated in a single API call. The model is prompted as an expert fact-checking system and asked to label each trace as SUPPORTS, REFUTES, or CONFLICTING based solely on the provided evidence. The response is required to be a JSON object with a single key `"verdicts"` containing an ordered array of labels, one per trace.

```
You are an expert fact-checking system.
Claim: "{claim}"
Evidence: "{evidence}"

Task: Evaluate if each of the following {N} Reasoning Traces logically
concludes that the claim is SUPPORTS, REFUTES, or CONFLICTING based ONLY
on the Evidence.

Traces:
Trace 0: {trace_0}
Trace 1: {trace_1} ...

Output ONLY a valid JSON object containing a single key called "verdicts",
which holds an array of your verdicts in the exact same order.
Example: {"verdicts": ["SUPPORTS", "REFUTES", "CONFLICTING", "SUPPORTS"]}
```

Probability construction. The raw batch verdicts are mapped to (*supports_weight*, *refutes_weight*, *conflicting_weight*) counts. To integrate these raw vote counts from the prompt variations into the continuous soft-voting ensemble, we apply a standard softmax function over the aggregated class frequencies to construct a pseudo-probability distribution $\mathbf{p}^{(3)}$:

$$\mathbf{p}^{(3)} = \mathrm{softmax}\big(\big[\, w_{\text{ref}},\ w_{\text{sup}},\ w_{\text{con}} \,\big]\big) \tag{3}$$

We explicitly acknowledge that applying a softmax function to discrete integer counts does not yield a calibrated probabilistic distribution. Because the exponential function heavily scales the distance between the majority count and the runners-up, this transformation purposefully yields an artificially sharp, high-confidence distribution. Rather than treating this as a calibrated probability equivalent to the internal logits of Approaches 1 and 2, we deploy this transformation as an aggressive heuristic prior. It forces the zero-shot pipeline to strongly dominate the weighted ensemble vote when its internal prompt consensus is high, while sharply suppressing its runner-up predictions.

Furthermore, a numerical consistency check is also run per trace: numbers extracted from the claim, evidence, and trace are compared pairwise, and traces whose numeric entities differ from the reference by more than 10% receive a consistency score of 0, dampening their weight in the vote. We adopted this 10% threshold as a strict heuristic safeguard after observing intermittent numerical hallucinations in GPT-4o-mini's outputs during early validation. While a formal ablation of this specific parameter was omitted due to API cost constraints, it serves as an essential architectural guardrail to prevent mathematically ungrounded traces from skewing the final ensemble.

4.5. Language-Specific Ensemble via Grid Search

Probability fusion. To construct the instance-level probability distributions $\mathbf{p}^{(1)}$ and $\mathbf{p}^{(2)} \in \Delta^2$ from the trace-level scores, we extract the maximum score assigned to each class across the entire trace pool and normalise them. Let T represent the set of reasoning traces for a given instance, and $C = \{\textit{false}, \textit{true}, \textit{conflicting}\}$ be the set of possible classes. For models $k \in \{1, 2\}$, the raw aggregated score $m_c^{(k)}$ for class $c \in C$ is defined as the maximum score among all traces predicting that class:

$$m_c^{(k)} = \max_{\{i \in T \mid v_i = c\}} s_i^{(k)} \tag{4}$$

If no trace predicts a particular class c, we set $m_c^{(k)} = 0$. These aggregated max-scores are then normalised to form a valid probability distribution over the three classes:

$$p_c^{(k)} = \frac{m_c^{(k)}}{\sum_{c' \in C} m_{c'}^{(k)}} \tag{5}$$

For Approach 3, the softmax-transformed batch weights (Equation 3) serve directly as the probability vector $\mathbf{p}^{(3)}$. The final blended ensemble distribution is computed via weighted soft-voting:

$$\mathbf{p}^{\text{ens}} = w_1\,\mathbf{p}^{(1)} + w_2\,\mathbf{p}^{(2)} + w_3\,\mathbf{p}^{(3)}, \quad \sum_{k=1}^{3} w_k = 1,\ w_k \geq 0 \tag{6}$$

The `Verdict_BoN` prediction is $\arg\max_c\, p_c^{\text{ens}}$.

Grid search optimisation. The weights (w_1, w_2, w_3) are selected independently for each language by exhaustive grid search on the official development set, maximising Macro Avg F1 on the `Verdict_BoN` output. The search sweeps $w_1, w_2 \in \{0.00, 0.05, 0.10, \ldots, 1.00\}$ and constrains $w_3 = 1 - w_1 - w_2 \geq 0$, yielding 231 candidate weight triples. The optimal weights found are shown in Table 2.

Table 2
Optimal ensemble weights per language, determined by grid search on the development set (Macro Avg F1 objective).

Language	w_1 (XLM-R)	w_2 (LLaMA)	w_3 (GPT-4o-mini)
English (EN)	0.00	0.60	0.40
Spanish (ES)	1.00	0.00	0.00
Arabic (AR)	1.00	0.00	0.00

The weight divergence is notable: English benefits entirely from generative decoder models (LLaMA + GPT-4o-mini), whereas Spanish and Arabic achieve their best development-set Macro F1 by relying exclusively on the fine-tuned XLM-RoBERTa encoder. We attribute this to the larger amount of English training data enabling stronger decoder fine-tuning, while XLM-RoBERTa's multilingual pre-training provides a more robust inductive bias for Spanish and Arabic with fewer labelled examples.

Blind test execution. For the hidden test set, the development-tuned weights in Table 2 are hardcoded directly into the submission scripts; no further grid search is performed, as ground-truth labels are unavailable at test time.

4.6. Post-Processing and Submission Alignment

ID force-alignment. During dataframe construction, dataset query IDs were cast as integers in some pipeline stages but stored as strings in the gold test files. A force-alignment step is applied at submission time to ensure that the `query_id` field in every output record exactly matches the type and format expected by the CodaLab evaluator, preventing "missing `line_no`" evaluation errors.

Dummy trace injection (Spanish and Arabic). The Spanish and Arabic ensemble scripts route their final submission through the XLM-RoBERTa output files (since $w_1 = 1.00$). A JSON parsing error in the Approach 1 inference pipeline for these two languages caused the textual content of the `Reasoning_traces` array to be lost: the field was present but contained empty strings rather than the original trace text.

The CodaLab evaluator enforces a strict structural requirement: the length of the `Reasoning_traces` array must exactly match the length of the `BoN_Verdict_list` array (typically 20 items per instance). An array length mismatch causes a crash that prevents any metric from being computed. To satisfy this constraint while preserving the integrity of the classification predictions, a post-processing script was applied that iterates over every instance in the Spanish

and Arabic submission files and overwrites the `Reasoning_traces` field with a list of structural placeholder strings of the correct length:

```
"Reasoning_traces": ["No reasoning trace provided."] × len(BoN_Verdict_list)
```

The `Verdict_BoN`, `BoN_Verdict_list`, and `score_list` fields were not modified by this script.

Impact on evaluation metrics. Because `Recall@k` and the composite `Average Score` are computed from the ranked trace strings, the injected placeholder traces render these IR metrics meaningless for Spanish and Arabic. However, the `Verdict_BoN` predictions — and therefore all *classification metrics* (`Macro Avg F1`, `True F1`, `False F1`, `Conflicting F1`) — are derived solely from the `Verdict_BoN` field, which remained completely unaltered. These scores are therefore **100% mathematically valid** and are reported as such in Section 5.

4.7. Computational Considerations

Our tri-model design incurs materially higher inference cost than a single-model baseline: for every instance, each of the up to 20 reasoning traces must be scored independently by XLM-RoBERTa, by the QLoRA-adapted LLaMA-3.2-1B decoder, and by a GPT-4o-mini API call, in addition to a per-trace BM25 retrieval step. This means roughly 3× the model invocations of a single-classifier system, plus non-trivial API latency and cost for Approach 3 (bounded by our concurrency limit of 3 and a 45-second per-request timeout, Section 4.4). We did not conduct a formal cost-benefit analysis (e.g., latency or dollar-cost per instance) against a single-model baseline such as XLM-RoBERTa alone, which on Spanish and Arabic development data already matches or exceeds the full ensemble (Table 3). Given that the marginal Macro F1 gain from the ensemble over the strongest single component is small for two of the three languages, we flag this cost-accuracy trade-off as a limitation and leave a systematic efficiency comparison to future work.

5. Results

We report results on two evaluation sets. **Development-set results** (Section 5.1) are drawn from our internal ablation experiments run against the official development partition and reflect the Macro Avg F1 of each individual component and the final ensemble. **Hidden test-set results** (Section 5.2) are the official CodaLab scores released after submission and are drawn exclusively from the `results and final rankings.txt` file.

A critical caveat applies to two of the three languages. As described in Section 4.6, a JSON parsing bug caused the `Reasoning_traces` field to contain only structural placeholder strings in the Spanish and Arabic submissions. Consequently:

- **Information-retrieval metrics** (`Recall@k`, `Average Score`) are *not reported* for Spanish and Arabic, as they are computed from trace text and are therefore invalid for those two languages.
- **Classification metrics** (`Macro Avg F1`, `True F1`, `False F1`, `Conflicting F1`) are derived solely from the `Verdict_BoN` field, which was never modified. These scores are fully valid for all three languages and are reported without qualification.

5.1. Development-Set Results

Table 3 reports the Macro Avg F1 on the official development set for each component model and for the final language-specific ensemble.

Several patterns emerge from Table 3. XLM-RoBERTa (Approach 1) provides the largest individual gains on Spanish and Arabic, while LLaMA (Approach 2) leads on English among the three fine-tuned

Table 3
Task 2 development-set Macro Avg F1 by approach and language. *Avg* is the unweighted mean across the three languages. SFT is supervised fine-tuning. Ensemble weights follow Table 2.

Approach	EN	ES	AR	Avg
Baseline	0.5362	0.3717	0.5364	0.4814
Approach 1: XLM-RoBERTa	0.5069	0.4541	**0.5603**	0.5071
Approach 2: LLaMA-3.2-1B SFT	0.5214	0.4184	0.5381	0.4926
Approach 3: GPT-4o-mini zero-shot	0.4618	0.3583	0.5025	0.4409
Ensemble (language-specific)	**0.5446**	**0.4541**	0.5588	**0.5192**

or prompted components. GPT-4o-mini zero-shot (Approach 3) consistently ranks last on Macro F1, though as noted in Section 4.5 its probability signal still complements the fine-tuned models in the English ensemble. We also note that the language-specific ensemble beats every individual approach on English and Spanish. For Arabic, the ensemble (0.5588) is marginally below XLM-RoBERTa alone (0.5603); this is discussed further in Section 6.2.

Table 4
Task 2 development-set per-class F1 for the final language-specific ensembles. † *Conflicting* F1 is 0.0000 for Arabic as no claims exist for this class in the development set.

Language	Macro F1	False F1	True F1	Confl. F1
English (EN)	0.5446	0.7000	0.4700	0.4600
Spanish (ES)	0.4541	0.8600	0.2100	0.2900
Arabic (AR)	0.5588	0.8700	0.8000	0.0000†

While Table 3 reports the Macro Avg F1 used as our primary optimization target during model selection, Table 4 provides the per-class F1 breakdown for the final language-specific ensembles on the development set. This breakdown reveals early indicators of the minority-class suppression that would later affect the hidden test set; for example, the Spanish ensemble's high *False* F1 (0.8600) was already masking poor performance on the *True* (0.2100) and *Conflicting* (0.2900) classes during development. A full comparison against the hidden test set is provided in Section 5.2.

Table 5
Task 2 development-set Recall@5 for evidence retrieval. *Avg* is the unweighted mean across the three languages. SFT is supervised fine-tuning.

Approach	EN	ES	AR	Avg
Baseline	0.3117	0.2346	0.2777	0.2747
Approach 1: XLM-RoBERTa	0.3172	**0.2857**	**0.2816**	**0.2948**
Approach 2: LLaMA-3.2-1B SFT	**0.3269**	0.2653	0.2789	0.2904
Approach 3: GPT-4o-mini zero-shot	N/A*	N/A*	N/A*	N/A*
Ensemble (language-specific)	0.3269	0.2857	0.2802	0.2976

*See discussion regarding the retrieval pipeline for Approach 3.

As shown in Table 5, the retrieval efficacy varies slightly across the fine-tuned models. Approaches 1 and 2 demonstrate modest improvements in Recall@5 over the raw baseline, as their scoring mechanisms successfully up-ranked relevant evidence snippets during the trace-generation phase.

The ensemble's Recall@5 inherits the trace ranking of whichever component receives the larger grid-search weight (Table 2): for English, it matches Approach 2, and for Spanish and Arabic, it tracks Approach 1. The Arabic ensemble recall of 0.2802 is marginally lower than the 0.2816 achieved by Approach 1 alone; this minor discrepancy arises from differing sample sizes after de-duplication (644 versus 652 claims), even though the underlying trace rankings remain identical.

Notably, Approach 3 reports N/A for Recall@5. This is a structural artifact of the zero-shot pipeline architecture. Because GPT-4o-mini (Approach 3) was constrained to act purely as an inference engine without a separate retriever module, it was evaluated by providing the concatenated, gold-standard evidence array directly from the development set JSON into the prompt context window, making empirical search metrics inapplicable.

5.2. Hidden Test-Set Results

Table 6 reports all valid classification metrics from the official hidden test-set evaluation. Table 7 reports the IR metrics that are available (English only). The official CLEF leaderboard ranks systems only by *Average Score*; the per-metric ranks and top-ranked scores reported below (for Macro F1, True F1, False F1, Conflicting F1, and Recall@5) are computed by us directly from the full per-team score sheet, so the top-ranked score in a given column may belong to a different team than in another column. We do not have access to competing teams' system descriptions or architectures, so the discussion below is limited to the magnitude of these gaps rather than their underlying cause.

Table 6
Official hidden test-set classification metrics, compared against the top-ranked score for each metric (see Section 5.2 for ranking methodology). Δ = Ours − Top-ranked. † *Conflicting* F1 not reported by the evaluator for Arabic as no claims exist for this class.

Language	Macro F1		True F1		False F1		Confl. F1	
	Score	Rank	Score	Rank	Score	Rank	Score	Rank
Top-ranked (EN)	0.6150	1/22	0.7400	1/22	0.8915	1/22	0.2439	1/22
Ours (EN)	0.5817	15/22	0.6482	21/22	**0.8915**	**1/22**	0.2055	7/22
Δ (EN)	−0.0333		−0.0918		0.0000		−0.0384	
Top-ranked (ES)	0.6749	1/16	0.6154	1/16	0.9131	1/16	0.5017	1/16
Ours (ES)	0.3031	16/16	0.0000	16/16	0.8657	12/16	0.0435	16/16
Δ (ES)	−0.3718		−0.6154		−0.0474		−0.4582	
Top-ranked (AR)	0.8679	1/15	0.8426	1/15	0.8933	1/15	—†	—
Ours (AR)	0.3557	15/15	0.4566	14/15	0.6105	14/15	—†	—
Δ (AR)	−0.5122		−0.3860		−0.2828		—	

Table 7
Official hidden test-set IR metrics, compared against the top-ranked score for each metric (see Section 5.2 for ranking methodology). Spanish and Arabic own-system IR scores are invalid due to the dummy-trace submission issue (see Section 4.6); top-ranked scores are shown for reference regardless. Δ = Ours − Top-ranked.

Language	Recall@5		Average Score	
	Score	Rank	Score	Rank
Top-ranked (EN)	0.2534	1/22	0.4286	1/22
Ours (EN)	0.2435	10/22	0.4126	12/22
Δ (EN)	−0.0099		−0.0160	
Top-ranked (ES)	0.3046	1/16	0.4847	1/16
Ours (ES)	*invalid — dummy traces*			
Top-ranked (AR)	0.3426	1/15	0.6043	1/15
Ours (AR)	*invalid — dummy traces*			

English. The ensemble achieves a Macro F1 of 0.5817, ranking 15th out of 22 teams, 0.0333 points behind the top-ranked score (0.6150). *False* F1 is exceptionally strong at 0.8915 (rank 1/22), reflecting the model's reliable identification of the dominant class. *True* F1 (0.6482) trails the top-ranked score

(0.7400) by 0.0918 points, the largest gap among English metrics, indicating that while the ensemble is more balanced than a pure majority-class predictor, minority class recall remains the primary weakness. *Conflicting* F1 (0.2055) trails by only 0.0384 points despite ranking 7th, suggesting the LLaMA + GPT-4o-mini decoder combination captures *Conflicting* evidence patterns competitively even without matching the leading score. On the IR side (Table 7), Recall@5 (0.2435, rank 10/22) trails the top-ranked score (0.2534) by only 0.0099 points, and Average Score (0.4126, rank 12/22) trails the top-ranked score (0.4286) by 0.0160 points. Both deltas are smaller than the Macro F1, Conflicting F1, and True F1 gaps — suggesting the ensemble's evidence-ranking quality sits closer to the leaderboard ceiling than its classification accuracy does.

Spanish. The ensemble achieves a Macro F1 of 0.3031, ranking last (16/16), 0.3718 points behind the top-ranked score (0.6749). *True* F1 collapses to 0.0000 against a top-ranked score of 0.6154 ($\Delta = -0.6154$), and *Conflicting* F1 to 0.0435 against 0.5017 ($\Delta = -0.4582$), while *False* F1 stays comparatively close to the top-ranked score (0.8657 vs. 0.9131, $\Delta = -0.0474$). This failure is analysed in detail in Section 6.1.

Arabic. The ensemble achieves a Macro F1 of 0.3557, ranking last among the fifteen Arabic submissions (15/15), 0.5122 points behind the top-ranked score (0.8679) — the largest absolute gap across all languages and metrics. *True* F1 (0.4566) and *False* F1 (0.6105) trail the top-ranked scores (0.8426, 0.8933) by 0.3860 and 0.2828 points respectively; both are non-trivial in absolute terms, suggesting the XLM-RoBERTa model learned meaningful signal from Arabic training data, but well below the ceiling achieved on the leaderboard. The absence of *Conflicting* F1 in the official evaluator output for Arabic is noted; we do not impute this value.

6. Error Analysis

6.1. Spanish: Distributional Shift and Minority-Class Collapse

The most striking failure in our submission is the complete collapse of *True* F1 to 0.0000 for Spanish, paired with a *Conflicting* F1 of only 0.0435. Because the hidden test set contained 254 *True* gold labels (Table 1), a *True* F1 of exactly zero indicates the model failed to correctly identify a single one, effectively defaulting to the majority *False* baseline.

Observed Performance Variance under Distributional Shift. The Spanish development and training sets were severely skewed, with the *False* class dominating at approximately 84.2%. To combat this, inverse-frequency class weights were applied during the XLM-RoBERTa fine-tuning phase. However, the hidden test set presented a significant distributional shift: it was notably *less* imbalanced than the training data, containing only 70.5% *False* instances and a much larger proportion of *True* instances (21.8%).

Factors Contributing to Minority-Class Suppression. Because the grid search assigned $w_1 = 1.00$ to XLM-RoBERTa for Spanish (Table 2), the full weight of the prediction fell on an encoder that had been heavily regularised to penalise minority-class errors in an 84% skewed environment. We hypothesise that this aggressive class-weighting caused the model to overfit to the specific lexical features of the few *True* examples present in the training set. When exposed to the broader, more diverse array of *True* claims in the hidden test set, the model failed to generalise. Consequently, the softmax threshold for the *True* class was rarely, if ever, crossed, causing the model to safely default to predicting *False* to minimise aggregate loss.

Conflicting F1. The residual *Conflicting* F1 of 0.0435 indicates the model made a small number of *Conflicting* predictions, but these were almost entirely incorrect or missed the true *Conflicting* instances.

This is consistent with *Conflicting* being the rarest class and sharing lexical features with both *True* and *False* claims, making it hardest for a surface-pattern classifier to isolate.

Remediation. Future work should explore dynamic threshold tuning rather than static loss-weighting during fine-tuning. Furthermore, this failure mode highlights a danger of grid-search ensemble optimisation: assigning 100% weight to a single encoder creates a brittle system. Fusion strategies that enforce a minimum weight bound on generative models (LLaMA, GPT-4o-mini) should be investigated, as generative architectures may be more resilient to distributional shifts than rigid sequence classifiers.

6.2. Arabic: Ensemble Weight Suboptimality

On the development set, XLM-RoBERTa alone achieves a Macro F1 of 0.5603 for Arabic, whereas the ensemble ($w_1 = 1.00$) achieves 0.5588. This negligible regression of 0.0015 is an artifact of the different pooling mechanisms used during evaluation. The standalone Approach 1 script derives its `Verdict_BoN` by taking the $\arg\max$ over the individual reasoning trace scores directly. In contrast, the ensemble pipeline (Section 4.5) aggregates predictions by extracting the maximum probability assigned to each class across the trace pool, normalising these three values into an instance-level distribution, and then applying an $\arg\max$. In rare edge cases where competing traces have nearly identical confidence scores, this structural difference in aggregation can flip the final predicted label.

Despite this minor pooling variance, the grid search assigned $w_1 = 1.00$ for Arabic because the pure-XLM-R configuration remained optimal at a 5% weight granularity. On the hidden test set, the Arabic Macro F1 is 0.3557 — substantially lower than the development-set figure. We attribute this drop primarily to the model overfitting to the development partition, compounded by a distributional shift in the hidden test data, though we did not run a separate train/dev/test calibration study to isolate the relative contribution of each factor. Furthermore, the performance gap between Arabic and English is consistent with the hypothesis that Arabic's morphological complexity poses additional challenges for multilingual XLM-RoBERTa pre-training; however, we did not run controlled experiments (e.g., morphological ablations or matched-data-size comparisons) to isolate this factor from confounds such as training set size and the complete absence of *Conflicting*-class examples in Arabic training data (Table 1).

6.3. English: Majority-Class Dominance and True Recall

While the English ensemble achieved a competitive Macro F1 of 0.5817 (rank 15/22), the per-class breakdown reveals a severe performance asymmetry that explains the rank disparity across metrics. The ensemble achieved a *False* F1 of 0.8915 (rank 1/22), demonstrating exceptional reliability and accuracy in identifying the dominant negative class. However, this success came at the direct expense of the *True* class, where the model's F1 score of 0.6482 ranked near the bottom (21/22). This paradox—a 15th place Macro F1 buoyed by a 1st place *False* F1 and dragged down by a 21st place *True* F1—implies that competing systems maintained far more balanced classification boundaries. Because *True* claims often feature direct lexical and numerical overlaps with the retrieved evidence, they are plausibly easier to verify, which may explain why most competing systems scored comparatively well on this class. We do not have access to competitors' architectures or training procedures, so we cannot confirm this directly; our own decoder-heavy ensemble appears to behave as a conservative verifier, sacrificing standard *True* predictions to hyper-optimise accuracy on the dominant *False* class.

Distributional Shift. As noted in Table 1, the proportion of *False* claims in the English track jumped from 57.1% in the development set to 70.3% in the hidden test set. Because the ensemble weights ($w_2 = 0.60$ for LLaMA, $w_3 = 0.40$ for GPT-4o-mini) were calibrated via grid search on the more balanced development set, the resulting fusion was not aggressive enough in predicting the minority classes under the heavily skewed test conditions. This is consistent with the models leaning on a learned

prior that a claim is likely *False* unless strong evidence proves otherwise, though we did not directly probe the models' internal confidence calibration to confirm this mechanism.

Conflicting Degradation. The performance on the *Conflicting* class dropped from an F1 of 0.4600 on the development set (Table 4) to 0.2055 on the hidden test set (Table 6), though it still secured a respectable rank of 7/22. This steep absolute drop is consistent with generative decoders (LLaMA and GPT-4o-mini) struggling to consistently identify epistemological conflict across different distributional splits, though confirming this would require targeted ablations isolating the decoders' behaviour on the *Conflicting* class specifically. The relatively high rank (7th) (Table 6) despite the low absolute score suggests that correctly classifying Conflicting claims was difficult for most systems in this year's English track, though we cannot generalise this observation beyond the teams that participated in this specific shared task

6.4. The JSON Trace Bug: Root Cause and Scope

The pipeline bug that caused empty `Reasoning_traces` arrays in the Spanish and Arabic submissions arose during the Approach 1 inference stage for those two languages. The `parse_trace` function uses a regular expression with the `DOTALL` flag to capture the `Justification` field between the literal strings `"Justification:"` and `"\n\nLabel:"`. In a subset of Spanish and Arabic traces, this delimiter pattern was absent or malformed — likely because the LLM that generated those traces used a slightly different formatting convention in non-English outputs. The regex therefore returned an empty string, and the downstream serialisation wrote an empty array to the JSON output file instead of the original trace strings.

The bug did not affect the `BoN_Verdict_list` or `score_list` fields, because those are populated from the normalised verdict and the model's softmax score respectively, neither of which depends on the regex-captured justification text. The `Verdict_BoN` field is similarly unaffected. As a result, all classification metrics remain fully valid, and only the trace-string-dependent IR metrics (`Recall@k`, `Average Score`) are invalidated for Spanish and Arabic.

The English submission was not affected because the English traces, generated by the same LLM template, consistently used the `"Justification: ... \n\nLabel:"` delimiter format, allowing the regex to capture the justification text correctly in all cases.

7. Conclusion

We presented a tri-model ensemble system for CLEF 2026 CheckThat! Task 2: Fact-Checking Numerical Claims. Our system combines a fine-tuned XLM-RoBERTa sequence classifier (Approach 1), a LLaMA-3.2-1B-Instruct generative reward model adapted with 4-bit QLoRA (Approach 2), and a zero-shot GPT-4o-mini scoring pipeline (Approach 3). Evidence for each reasoning trace is grounded via BM25Okapi Top-1 retrieval, and the three probability distributions are fused using language-specific weights determined by exhaustive grid search on the official development set.

On the hidden test set, the ensemble achieved a Macro Avg F1 of **0.5817** for English (rank 15/22), **0.3031** for Spanish (rank 16/16), and **0.3557** for Arabic (rank 15/15). The English submission was fully intact and produced a *False* F1 of 0.8915 — the highest among all 22 English systems — demonstrating strong identification of the dominant class. However, the Spanish and Arabic submissions were affected by a JSON parsing bug that caused the `Reasoning_traces` field to contain only structural placeholder strings, invalidating information-retrieval metrics (`Recall@k`, `Average Score`) for those two languages while leaving all classification metrics mathematically valid.

The core lessons from this work are threefold. First, language-specific ensemble weights are essential: the optimal mixture for English (decoder- heavy, with LLaMA at 60% and GPT-4o-mini at 40%) is entirely different from Spanish and Arabic (encoder-only, with XLM-RoBERTa at 100%), reflecting the different strengths of bidirectional versus autoregressive architectures across languages and data scales.

Second, class-weight correction alone is insufficient to overcome extreme label skew: the Spanish *True* F1 collapse to 0.0000 on the hidden test set shows that assigning the full ensemble weight to the encoder under high class imbalance can cause catastrophic minority-class suppression. Third, submission pipeline robustness is as important as model quality: a single regex failure in trace parsing propagated silently through the serialisation stage and invalidated IR metrics for two out of three languages.

Future work. For Task 2 specifically, we identify eight concrete directions. *(i) Data augmentation and calibration:* to address the minority-class suppression observed in Spanish and Arabic, future work will explore oversampling or back-translation for *True* and *Conflicting* instances, alongside re-estimating class weights from test-distribution priors and enforcing per-class F1 floor constraints to prevent minority-class suppression. *(ii) Robust trace serialisation:* a language-agnostic fallback in the trace parser—preserving the raw trace string when the regex fails, rather than returning an empty field—would have prevented the dummy-trace workaround and preserved IR metric validity. *(iii) Larger decoder models:* replacing LLaMA-3.2-1B with a larger instruction-tuned decoder—subject to GPU memory constraints—may improve the generative reward signal and provide a stronger complement to the encoder in multilingual settings. *(iv) Systematic efficiency evaluation:* as noted in Section 4.7, our ensemble prioritises accuracy over inference speed; future work will conduct a formal cost-benefit analysis, including dollar-cost and latency metrics, to better align our tri-model design with resource-constrained production environments. *(v) Controlled analysis of performance gaps:* as discussed in Section 6, several of our findings regarding Arabic morphological bottlenecks and model behaviour on the *Conflicting* class remain speculative. Future work will include controlled ablations—such as morphological testing and targeted decoder probing—to isolate these factors from potential confounds like training data imbalance, set-size variance, and the unobserved architectural choices of competing systems. Furthermore, while we attribute the ensemble's tendency toward *False* predictions to a learned prior, we plan to directly probe internal confidence calibration to confirm whether this behaviour is a result of prior bias or insufficient test-time verification signal. *(vi) Alignment-based prompting for Approach 2:* the current prompt (Section 4.3) asks the LLaMA reward model to re-derive a verdict from the claim, evidence, and trace justification — effectively repeating the fact-checking task the trace has already performed. Since each reasoning trace is itself the output of a pretrained LLM's fact-checking pass, a more principled formulation is to ask the model whether the trace's justification and verdict are *mutually consistent* with the claim and evidence, i.e., an alignment or entailment judgment rather than an independent re-verification. Future work will reformulate the prompt and next-token scoring around this alignment objective and compare it against the current verdict-prediction formulation. *(vii) Comparison against test-time scaling baselines:* our evaluation reports the tri-model ensemble in isolation; future work will benchmark it against simpler test-time scaling baselines applied to the same reasoning traces, such as unweighted majority voting over trace verdicts and self-consistency voting, to quantify the specific contribution of the learned scoring components over naive aggregation. *(viii) Generalisation to public non-English test sets:* the hidden test-set results in Section 5.2 are limited to the official CheckThat! evaluation. Future work will evaluate the trained tri-model ensemble on publicly available Spanish and Arabic fact-checking benchmarks to assess whether the performance gaps observed here are specific to this shared task's test distribution or reflect a more general weakness in non-English numerical claim verification.

Acknowledgments

We thank the CLEF 2026 CheckThat! Lab organisers for designing the tasks, providing the datasets, and operating the CodaLab evaluation infrastructure. We would also like to thank the faculty and teaching staff of CS/CE 335/466 (Introduction to Large Language Models) at Habib University for their guidance throughout this project. Experiments were run on Kaggle GPU notebooks (T4 hardware) made freely available through the Kaggle research programme.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) in order to: assist with drafting and proofreading of the manuscript. The authors also used GPT-4o-mini (OpenAI, via OpenRouter) as a zero-shot scoring component within the experimental system described in Section 4.4; this is a methodological component of the described system, not a writing aid, and is fully documented there. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] E. S. Salido, A. Barrón-Cedeño, Alba García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 working notes, CLEF 2026, Jena, Germany, 2026.

[2] J. M. Struß, S. Schellhammer, S. Dietze, V. V., V. Setty, T. Chakraborty, P. Nakov, A. Anand, P. Chungkham, S. Hafid, D. Sahnan, K. Todorov, Overview of the CLEF-2026 CheckThat! Lab: Advancing multilingual fact-checking, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. S. Salido, A. Barrón-Cedeño, Alba García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer, 2026.

[3] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, V. Setty, M. Sundriyal, K. Todorov, V. V., The CLEF-2025 CheckThat! Lab: Subjectivity, fact-checking, claim normalization, and retrieval, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 467–478. doi:10.1007/978-3-031-88720-8_68.

[4] P. Nakov, A. Barrón-Cedeño, T. Elsayed, R. Suwaileh, L. Màrquez, W. Zaghouani, P. Atanasova, S. Kyuchukov, G. Da San Martino, Overview of the CLEF-2018 CheckThat! Lab on automatic identification and verification of political claims, in: P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J. Y. Nie, L. Soulier, E. SanJuan, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer International Publishing, Cham, 2018, pp. 372–387. doi:10.1007/978-3-319-98932-7_32.

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

[6] Y. Ji, J. Li, Y. Xiang, H. Ye, K. Wu, K. Yao, J. Xu, L. Mo, M. Zhang, A survey of test-time compute: From intuitive inference to deliberate reasoning, Computational Linguistics (2026) 1–51. URL: https://doi.org/10.1162/COLI.a.629. doi:10.1162/COLI.a.629.

[7] E. Zhao, P. Awasthi, S. Gollapudi, Sample, scrutinize and scale: Effective inference-time search by scaling verification, in: Proceedings of the 42nd International Conference on Machine Learning, volume 267, PMLR, 2025, pp. 77272–77309. URL: https://proceedings.mlr.press/v267/zhao25a.html.

[8] Y. Wang, P. Zhang, S. Huang, B. Yang, Z. Zhang, F. Huang, R. Wang, Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding, in: Advances in Neural Information Processing Systems, 2025. URL: https://proceedings.neurips.cc/paper_files/paper/2025/hash/ed45d6a03de84cc650cae0655f699356-Abstract-Conference.html.

[9] 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: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. doi:10.18653/v1/2020.acl-main.747.

[10] A. S. Anik, M. F. K. Chowdhury, A. Wyckoff, S. R. Choudhury, ClaimIQ at CheckThat! 2025: Comparing prompted and fine-tuned language models for verifying numerical claims, in: Working Notes of CLEF 2025 — Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 795–803. URL: https://ceur-ws.org/Vol-4038/paper_61.pdf.

[11] V. Venktesh, et al., Overview of the CLEF-2025 CheckThat! Lab task 3 on fact-checking numerical claims, in: Working Notes of CLEF 2025 — Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 709–721. URL: https://ceur-ws.org/Vol-4038/paper_53.pdf.

[12] M. Sawiński, K. Węcel, E. Księżniak, M. Stróżyna, W. Lewoniewski, P. Stolarski, W. Abramowicz, OpenFact at CheckThat! 2023: Head-to-head GPT vs. BERT - a comparative study of transformers language models for the detection of check-worthy claims, in: Working Notes of CLEF 2023 — Conference and Labs of the Evaluation Forum, volume 3497 of *CEUR Workshop Proceedings*, 2023, pp. 453–472. URL: https://ceur-ws.org/Vol-3497/paper-040.pdf.

[13] F. A. Leistra, T. Caselli, Thesis titan at CheckThat! 2023: Language-specific fine-tuning of mDeBERTaV3 for subjectivity detection, in: Working Notes of CLEF 2023 — Conference and Labs of the Evaluation Forum, volume 3497 of *CEUR Workshop Proceedings*, 2023, pp. 351–359. URL: https://ceur-ws.org/Vol-3497/paper-030.pdf.

[14] S. Gruman, L. Kosseim, CLaC-2 at CheckThat! 2024: A zero-shot model for check-worthiness and subjectivity classification, in: Working Notes of CLEF 2024 — Conference and Labs of the Evaluation Forum, volume 3740 of *CEUR Workshop Proceedings*, 2024, pp. 446–452. URL: https://ceur-ws.org/Vol-3740/paper-41.pdf.

[15] S. Adhikari, H. Sharma, R. Kumari, S. Satapara, M. S. Desarkar, DEFAULT at CheckThat! 2024: Retrieval augmented classification using differentiable top-K operator for rumor verification based on evidence from authorities, in: Working Notes of CLEF 2024 — Conference and Labs of the Evaluation Forum, volume 3740 of *CEUR Workshop Proceedings*, 2024, pp. 351–360. URL: https://ceur-ws.org/Vol-3740/paper-30.pdf.

[16] M. Staudinger, A. El-Ebshihy, W. Kusa, F. Piroi, A. Hanbury, ATOM at CheckThat! 2025: Retrieve the implicit - scientific evidence retrieval, in: Working Notes of CLEF 2025 — Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 1246–1255. URL: https://ceur-ws.org/Vol-4038/paper_98.pdf.

[17] M. E. Bulut, K. E. Keleş, M. Kutlu, TurQUaz at CheckThat! 2024: A hybrid approach of fine-tuning and in-context learning for check-worthiness estimation, in: Working Notes of CLEF 2024 — Conference and Labs of the Evaluation Forum, volume 3740 of *CEUR Workshop Proceedings*, 2024, pp. 369–377. URL: https://ceur-ws.org/Vol-3740/paper-32.pdf.

[18] Meta AI, Introducing Llama 3.2, https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/, 2024. Accessed: 2026-05-27.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: Proceedings of the 10th International Conference on Learning Representations (ICLR 2022), 2022. URL: https://openreview.net/pdf?id=nZeVKeeFYf9.