# GippLab at CheckThat! 2026: Multilingual Numerical Claim Verification with QLoRA Trace Scoring

CheckThat at CLEF 2026

Ahmad Raza Khawaja[1,*], Adam Moshe Lehavi[1]

[1]University of Göttingen, Wilhelmsplatz 1, 37073 Göttingen, Germany

**Abstract**

This paper describes the GippLab submission to CheckThat! 2026 Task 2 on multilingual numerical claim verification. The task requires ranking candidate reasoning traces and predicting a final claim verdict for English, Spanish, and Arabic claims. We explored three main approaches: zero-shot prompted ranking with a frozen instruction-tuned language model, preference-based ranking with Direct Preference Optimization (DPO), and supervised trace scoring. The prompted baselines used listwise and pairwise ranking prompts, but were sensitive to the order of candidate traces. We observed considerable positional bias in both the prompt-based model and the DPO model. Our best system fine-tunes `Llama-3.1-8B` with Quantized LoRA as a multilingual trace scorer. It scores each claim-trace pair independently, ranks traces by their predicted relevance, and derives the final verdict from the top-ranked traces. This supervised trace-scoring formulation gave the best results among our experiments and was used for the final submission. Across the English, Spanish, and Arabic validation sets, the final trace-scoring approach achieved a mean macro-F1/Recall@5 score of 0.4350, beating the prompting baseline by 22.8% and the English DPO model by 17.7%. This approach scored first in the English category, achieving a mean macro-F1/Recall@5 score of 0.4286.

**Keywords**

Multilingual fact-checking, Numerical claim verification, Reasoning trace ranking, Trace scoring, QLoRA, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, Numeric-aware modeling, Long-context evidence, Direct Preference Optimization

## 1. Introduction

Numerical claims are frequent in political discourse, news, and social media. They often involve quantities, dates, percentages, rankings, or comparisons, where small changes in scope or numerical interpretation can alter the verdict drastically [1]. This makes numerical claim verification more demanding than standard textual entailment, especially in multilingual settings where systems must preserve numerical meaning across languages [2].

CheckThat! 2026 Task 2 focuses on multilingual numerical claim verification for English, Spanish, and Arabic [3]. This setup frames fact verification as a verifier-guided reranking problem, where systems must identify the most useful reasoning traces and use them to predict the final claim verdict. Each example provides a claim, evidence snippets, candidate reasoning traces, and trace-level verdicts. The system must rank the candidate traces and predict the final claim-level verdict. This setup is useful because the traces provide candidate explanations, but it also introduces a ranking challenge where the system must identify which traces are most reliable with respect to the claim and evidence.

We studied this task as a ranking problem. We first tested frozen prompt-based baselines, including listwise prompting and pairwise prompting, where traces are compared two at a time and the final ranking is built from the accumulated preferences [4]. However, Positional bias, defined as the model's tendency to rank based on prompt position and numerical ordering instead of content, had a considerable affect on our outputs [5]. We then trained a DPO ranking model on an expanded English dataset built

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

*Corresponding author.

ahmad.khawaja@stud.uni-goettingen.de (A. R. Khawaja); adam.lehavi@uni-goettingen.de (A. M. Lehavi)

https://github.com/ahmadrazakhawaja (A. R. Khawaja); https://github.com/aLehav (A. M. Lehavi)

with Google Translate [6]. Finally, we moved to supervised trace scoring, where each claim-trace pair is scored independently and the ranking is obtained directly from learned trace-level scores.

Our submitted system fine-tunes `Llama-3.1-8B-Instruct` with QLoRA on combined English, Spanish, and Arabic data [7, 8]. The main contribution of this working note is a practical comparison of ranking formulations for multilingual numerical claim verification, showing why supervised trace scoring was more controllable than generative ranking for our setting. On the official leaderboard, the submitted trace-scoring system ranked first on English, third on Spanish, and sixth on Arabic, indicating that supervised trace scoring was especially effective for the English setting while remaining competitive across the multilingual task.

## 2. Task and Dataset

The task follows a test-time scaling setup for fact verification [9]. For each claim and its evidence, multiple reasoning traces are generated by an LLM using different temperature values to induce diverse reasoning paths. After removing redundant traces, participants are asked to train a verifier model that ranks the remaining traces according to their utility for reaching the correct verdict. The final claim verdict is then derived from the top-ranked traces. Thus, the task evaluates both the quality of trace ranking and the correctness of the predicted claim verdict.

The Task already provides multilingual numerical claim verification examples in English, Spanish, and Arabic. We use the English dataset released in QuanTemp, as well as the 3260 Arabic and 2808 Spanish claims from the previous edition of CLEF 2025 with corresponding evidence collection [10]. Each example contains a claim, evidence snippets, candidate reasoning traces, a verdict for each trace, and a gold claim-level label. The claim-level labels are `True`, `False`, and `Conflicting`.

The system output has two components: a ranking of the candidate reasoning traces and a final claim verdict. The official evaluation is performed separately for each language. Trace-ranking quality is measured using Recall@5 over traces whose verdict matches the gold claim label, while claim-verdict prediction is evaluated using macro-F1. The final score for each language is the average of Recall@5 and macro-F1.

The system output has two components: a ranking of the 15 candidate reasoning traces and a final claim verdict. For a claim $c$, Recall@5 is computed as

$$\text{Recall@5}(c) = \frac{\left|\{i \in r_{1:5}^{(c)} : v_i = y_c^*\}\right|}{\left|\{i \in R^{(c)} : v_i = y_c^*\}\right|}, \tag{1}$$

where $r_{1:5}^{(c)}$ denotes the top five ranked traces for claim $c$, $R^{(c)}$ is the full set of candidate traces, $v_i$ is the verdict of trace $i$, and $y_c^*$ is the gold claim label.

A limitation of the dataset was that some claims had no candidate trace whose verdict matched the gold claim label, which made it impossible for any ranking model to retrieve a correct-verdict trace for those examples and lowered the maximum achievable Recall@5. Recall@5 is upper-bounded by the candidate trace set. For a claim with $m$ gold-matching traces, even a perfect ranking can retrieve at most five of them, giving a maximum claim-level Recall@5 of $\min(5, m)/m$. If $m = 0$, the evaluator assigns Recall@5 of 0.0 for that claim, since no candidate trace has a verdict matching the gold label. We therefore computed a Recall@5 upper bound on the validation set, obtaining 0.39 for English, 0.32 for Spanish, and 0.28 for Arabic. On the official test set, the corresponding Recall@5 upper bounds were 0.28 for English, 0.32 for Spanish, and 0.34 for Arabic.

## 3. Related Work

Large language models have been widely studied as zero-shot and few-shot rankers for natural language processing tasks. In listwise ranking, the model receives multiple candidates in a single prompt and is asked to output an ordered ranking [11]. This setting is attractive because it can be applied without

task-specific training, but prior work has shown that LLM rankings can be sensitive to prompt format, candidate order, and other presentation effects [12].

Pairwise ranking is another common formulation, where the model compares two candidates at a time and the final ranking is obtained by aggregating pairwise preferences [4]. This reduces the complexity of each individual decision compared with listwise ranking, but it introduces runtime constraints because the number of comparisons grows quadratically with the number of candidates. For $n$ traces, one-direction pairwise ranking requires $n(n-1)/2$ comparisons, and bidirectional ranking requires $n(n-1)$ comparisons. Recent work has also studied systematic biases in LLM-as-judge settings, including order and position effects in pairwise comparisons [5].

Preference optimization methods provide a supervised alternative for aligning model outputs with human or automatically constructed preferences. Direct Preference Optimization (DPO) trains a policy model to assign higher likelihood to preferred outputs than to rejected outputs, while regularizing against a reference model [13]. DPO has become a common method for preference-based fine-tuning because it avoids explicit reward modeling and reinforcement learning.

Parameter-efficient fine-tuning methods are widely used to adapt large language models under limited computational resources. Low-Rank Adaptation (LoRA) updates a small set of trainable low-rank matrices while keeping the base model weights frozen, reducing memory and training cost compared with full fine-tuning [14]. Quantized LoRA (QLoRA) further reduces memory usage by loading the base model in low precision while training LoRA adapters on top of it [7]. Instruction-tuned open-weight models such as `Mistral-7B-Instruct` have also been commonly used as strong baselines for prompting and adapter-based fine-tuning because they provide competitive instruction-following behavior at a manageable model size [15]. Additionally we used PyTorch scaled dot-product attention (SDPA), which computes the standard Transformer attention operation $\mathrm{softmax}(QK^\top/\sqrt{d_k})V$ with optimized PyTorch kernels to improve memory efficiency and runtime for long-context inputs without changing the model architecture [16].

## 4. Method

We explored the task through a sequence of increasingly supervised ranking approaches. We first evaluated frozen prompted baselines, then moved to preference-based DPO training, and finally trained supervised multilingual trace scorers. Across all approaches, the central goal was to rank the candidate reasoning traces for each claim and derive a final claim verdict from the highest-ranked traces.

### 4.1. Zero-Shot Listwise Prompted Ranking

Our first baseline used a frozen instruction-tuned language model without task-specific fine-tuning. As shown in Figure 1, the model received the claim, evidence snippets, and all candidate reasoning traces in a single prompt. It was instructed to rank all traces from best to worst and to predict the final claim verdict. The expected output was a strict JSON object containing a permutation of trace indices and one predicted verdict.

This listwise setup was simple and required no training, but the results were limited. In particular, we observed a strong positional bias. Even though the prompt explicitly instructed the model not to copy the input order unless it was truly the best ranking, the generated rankings were still sensitive to the original trace order. Because of this, the listwise prompting baseline did not perform well enough for the final system and motivated our later pairwise and supervised trace-scoring experiments.

### 4.2. Pairwise Prompted Ranking

We introduced pairwise prompted ranking to reduce the positional bias observed in the listwise prompting setup. In the listwise setting, the model received all candidate traces in one prompt, and we observed that the produced ranking could be influenced by the original order of the traces. To make the compari-

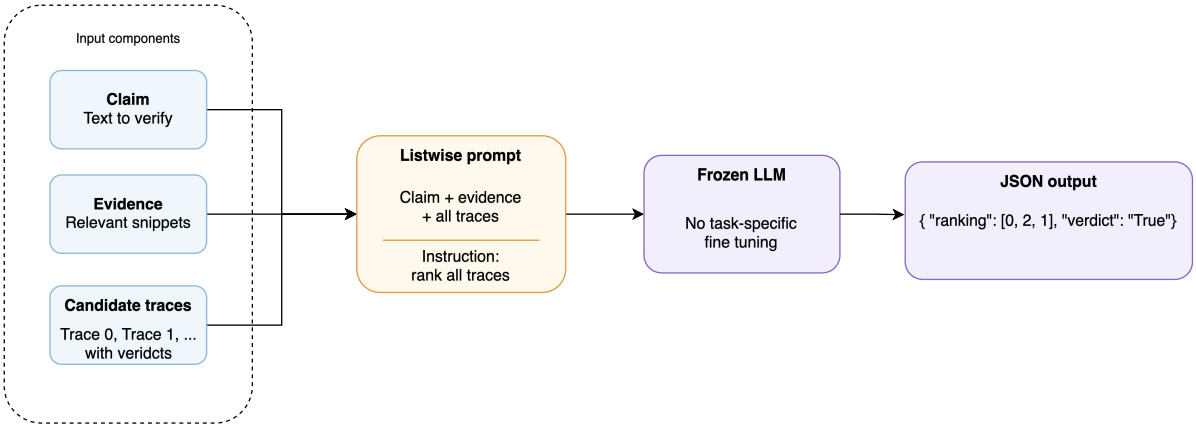


**Figure 1:** Listwise prompted ranking setup. A frozen language model receives the claim, evidence, and all candidate traces in one prompt, then generates a complete ranking and final verdict.

son more local and less dependent on the full list order, we reformulated ranking as a set of pairwise decisions.

In the pairwise setting, illustrated in Figure 2, the model was shown two candidate traces at a time and asked to select the more reliable one with respect to the claim and evidence. Each pairwise win contributed to the selected trace's score, and the final ranking was obtained by sorting traces according to their accumulated wins.

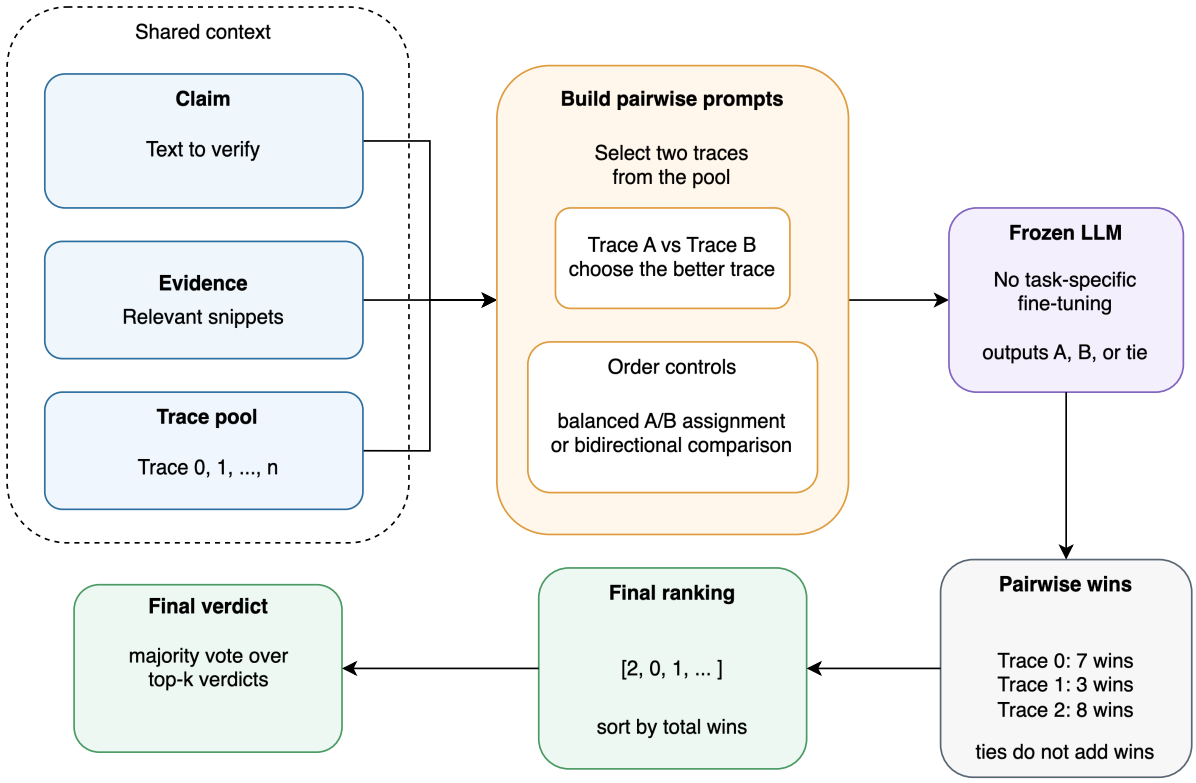


**Figure 2:** Pairwise prompted ranking setup. A frozen language model compares two traces at a time, and accumulated pairwise wins are used to produce the final ranking.

We tested two strategies to further reduce A/B positional bias. In the balanced setting, the assignment of traces to Trace A and Trace B was alternated, so that the lower-indexed trace was not always shown first. In the bidirectional setting, each pair was evaluated twice, once in each order. A win was counted only when both directions agreed; otherwise, the comparison was treated as a tie. These strategies

reduced some order effects, but did not fully remove positional bias, which motivated us to explore DPO-based ranking as a more supervised alternative.

### 4.3. DPO Ranking Mode

After the prompted baselines, we trained an English-only DPO/LoRA ranking model based on `Mistral-7B-Instruct-v0.3`. The DPO ranking setup is illustrated in Figure 4. The DPO training data was created from the expanded English dataset, which combined original English examples with Google Translate translations of Arabic and Spanish examples into English. The complete DPO pipeline, including translation, preference-pair construction, DPO fine-tuning, and cross-lingual evaluation, is shown in Figure 3.

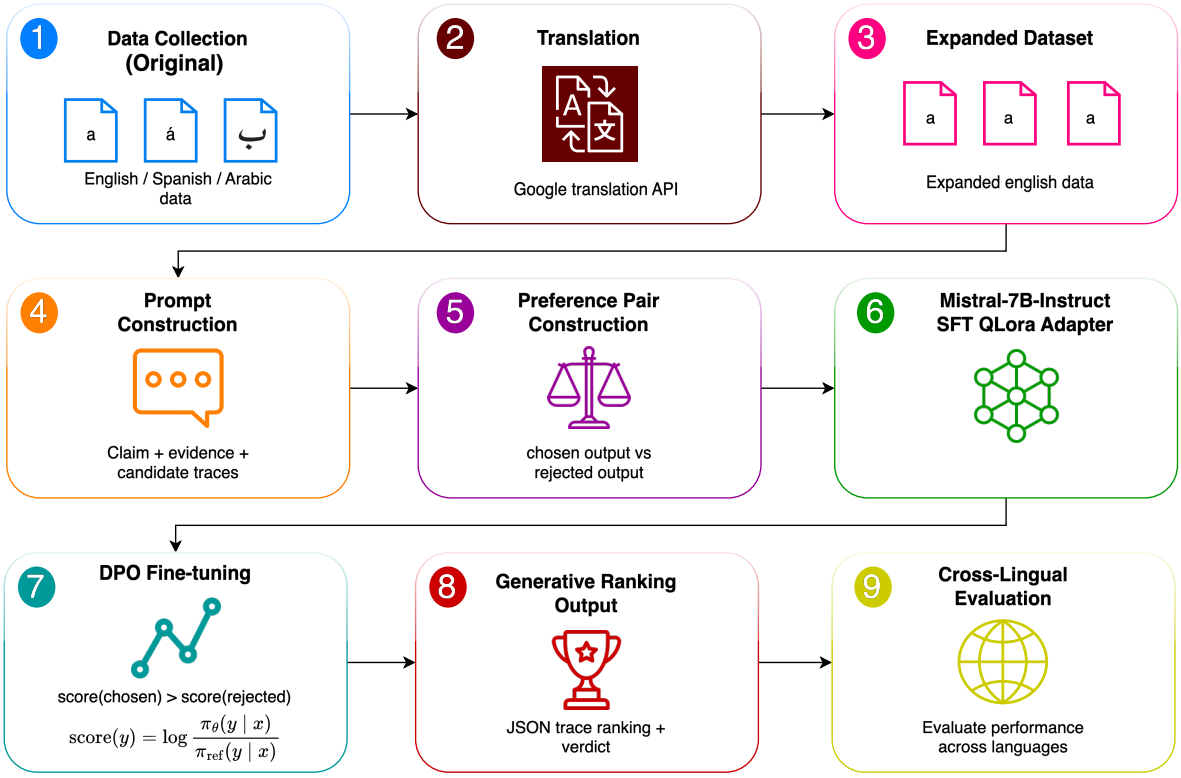


**Figure 3:** DPO ranking pipeline. Translated multilingual data is converted into ranking prompts and chosen/rejected preference pairs for DPO fine-tuning, followed by cross-lingual evaluation.

For each claim, we constructed a preferred output and one or more rejected outputs. The preferred output ranked traces whose verdict matched the gold claim label before traces whose verdict did not match the gold label, and predicted the gold verdict. Rejected outputs were designed as hard negatives, including rankings that placed a non-matching trace first, rankings with the top traces swapped, outputs with an incorrect predicted verdict, or combinations of incorrect ranking and incorrect verdict. The goal was to teach the model to prefer outputs that ranked label-consistent traces higher and predicted the correct claim verdict.

Although DPO improved over purely zero-shot prompting in some settings, it still inherited limitations from the generative ranking formulation. The model had to generate a complete JSON ranking, so the output remained sensitive to prompt formatting and trace order, and positional bias was still present. In addition, the DPO objective optimizes preference between complete generated outputs rather than directly optimizing per-trace relevance scores or the official ranking metric. This mismatch made it difficult to control trace-level ranking behavior reliably. For these reasons, we moved to supervised trace scoring, where each claim-trace pair is scored independently and the ranking is produced directly from learned trace-level scores.

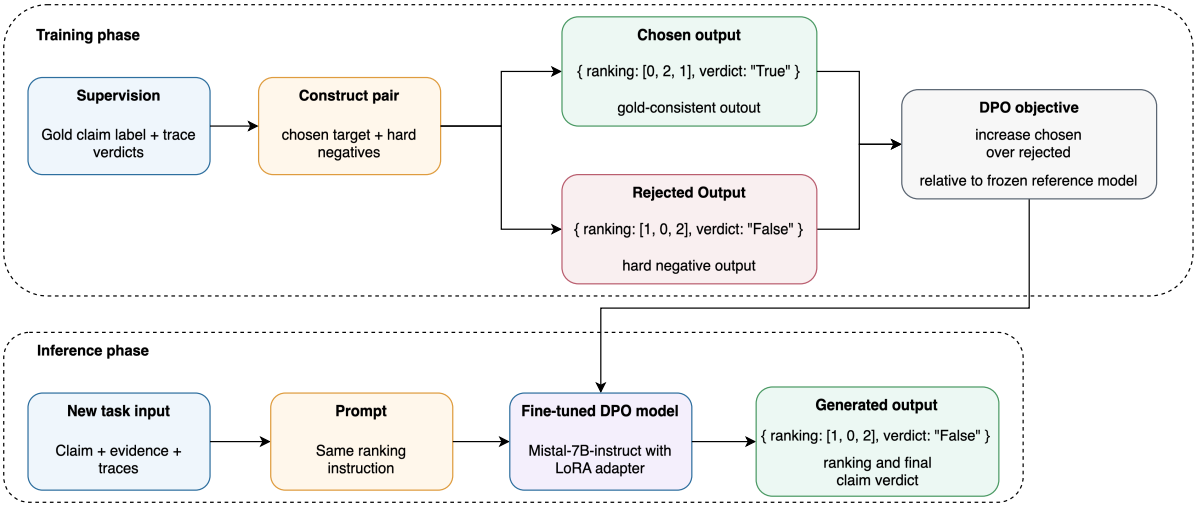


**Figure 4:** DPO ranking setup. Chosen and rejected JSON outputs are automatically constructed from the gold label and trace verdicts, then used to train the model to prefer gold-consistent rankings.

## 4.4. Supervised Trace Scoring

Our main submitted approach uses supervised trace scoring rather than full ranking generation. The input-output flow of the supervised trace scorer is shown in Figure 5. For each claim-trace pair, we construct a binary classification example. A trace is labeled positive if its trace verdict matches the gold claim label and negative otherwise. This converts the ranking task into a supervised trace-classification problem.

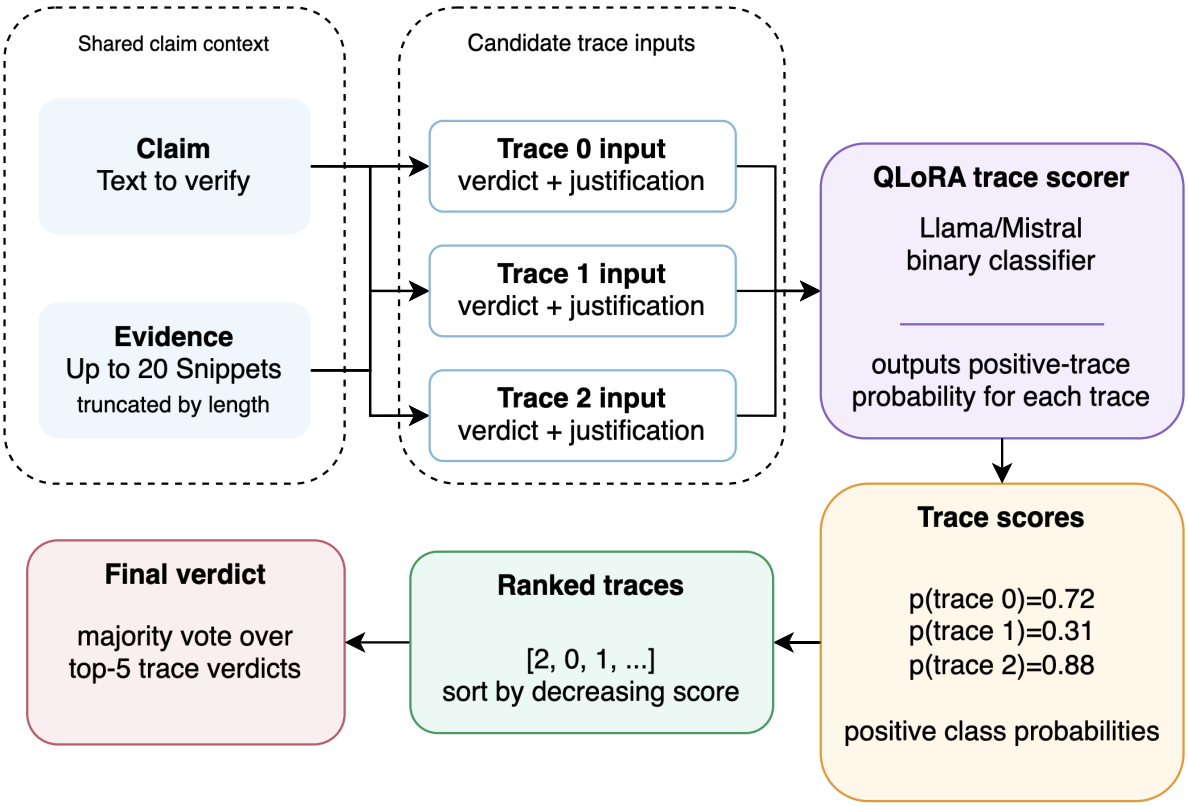


**Figure 5:** Input-output flow of the supervised trace scorer. Candidate traces are scored independently, sorted by positive-class probability, and aggregated through the top-ranked trace verdicts.

The input to the scorer contains the claim, evidence snippets, the trace verdict, and the cleaned trace justification:

```
Claim: <claim text>
Evidence snippets:
[0] <evidence snippet>
[1] <evidence snippet>
...
Verdict: <trace verdict>
Justification: <cleaned trace justification>
```

The submitted system fine-tunes `Llama-3.1-8B-Instruct` as a multilingual binary trace scorer using QLoRA with 4-bit loading. The model is trained on combined English, Spanish, and Arabic data. Most trace-scorer experiments use a two-logit cross-entropy classification head, while we also tested a one-logit binary cross-entropy variant and regularized some runs using LoRA dropout and weight decay. During inference, the positive-class score is used as the trace reliability score.

For a claim $c$, let $R^{(c)} = \{1, \ldots, n_c\}$ denote the set of candidate reasoning traces, where $n_c$ is the number of traces for that claim. Each trace $i \in R^{(c)}$ has a trace verdict $v_i$, and the claim has a gold label $y_c^*$. We build one scorer input $x_i$ for each trace by concatenating the claim text, evidence snippets, trace verdict, and trace justification.

The binary training label for trace $i$ is defined as

$$z_i = \begin{cases} 1, & \text{if } v_i = y_c^*, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let $f_\theta$ denote the fine-tuned trace scorer with parameters $\theta$. Given $x_i$, the model outputs two logits, one for the negative class and one for the positive class. We apply a softmax over the negative and positive logits and use the resulting positive-class probability as the trace score:

$$p_i = \mathrm{softmax}(f_\theta(x_i))_{\mathrm{pos}} \tag{3}$$

The model is trained with cross-entropy loss using the binary labels $z_i$. At inference time, each trace is scored independently, and the traces are ranked in decreasing order of $p_i$:

$$r^{(c)} = \mathrm{argsort}_{i \in R^{(c)}}(-p_i) \tag{4}$$

The final claim verdict is derived by majority vote over the verdicts of the top five ranked traces:

$$\hat{y}_c = \mathrm{majority}\left(\{v_i \mid i \in r_{1:5}^{(c)}\}\right) \tag{5}$$

### 4.5. Numeric-Aware and Evidence Ablations

Because the task focuses on numerical claims, we also tested numeric-aware variants. One variant used numeric embeddings, where numerical spans were detected and represented with additional numeric features. Another variant appended normalized number hints extracted from the claim, evidence, and trace text. These hints were intended to make numerical comparisons more explicit for the model.

As illustrated in Figure 6, in the numerical embeddings setting, numerical spans in the claim, evidence, and trace text were detected and canonicalized before tokenization. The canonicalization step normalized different number formats, including comma and decimal separators, percentages, and Arabic/Persian digits. Each detected number was marked in the text with a special `<num>` token while keeping the original surface form. For each `<num>` token, the canonical numeric value was encoded as a sequence of numeric characters and passed through a small GRU-based value encoder. The resulting vector replaced the standard token embedding at the `<num>` position, giving the model a value-aware representation of numerical expressions. This approach was intended to make numerical comparisons more robust than relying only on subword tokenization [17].

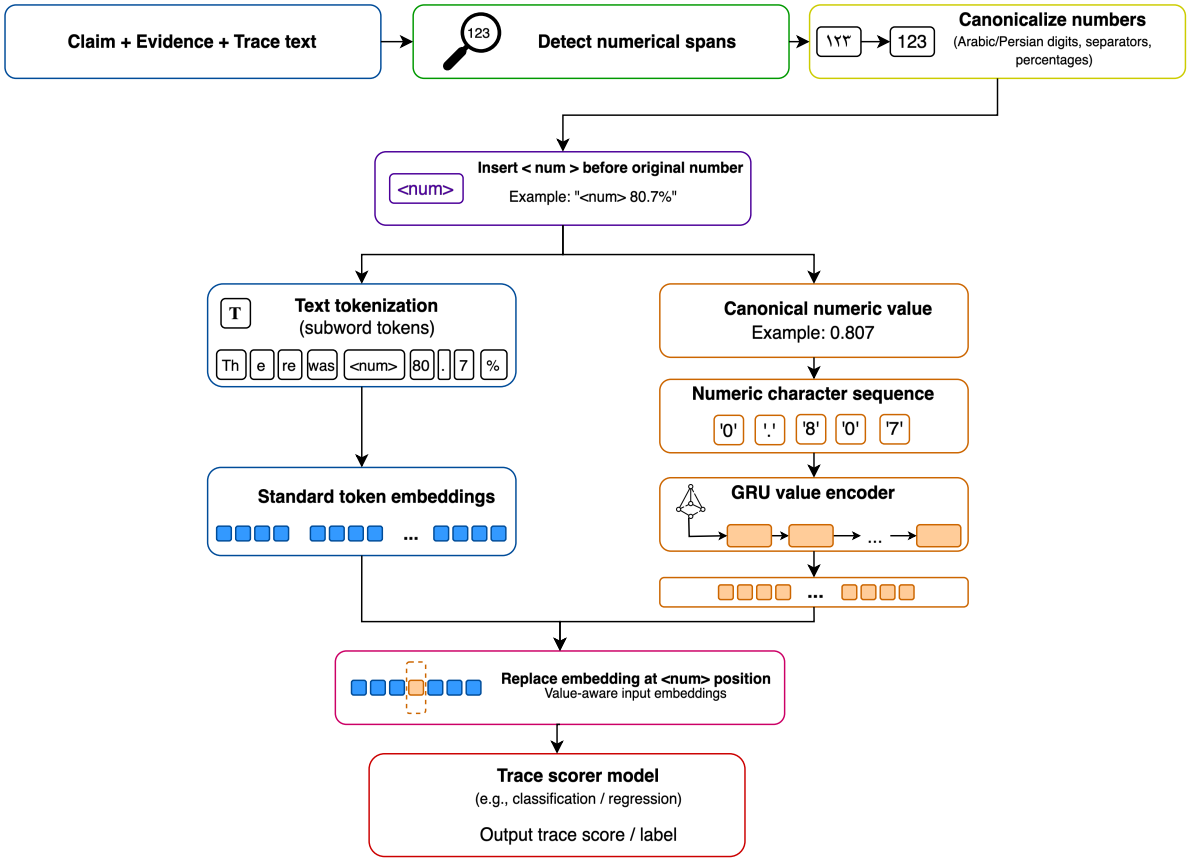


**Figure 6:** Numeric embedding pipeline. Numerical spans are canonicalized, marked with <num>, encoded with a GRU value encoder, and inserted as value-aware embeddings before trace scoring.

In a separate numeric-normalization variant, we appended plain-text normalized number hints extracted from the claim, evidence, and trace. Unlike numeric embeddings, which modify the embedding representation at <num> positions, this variant exposed canonical numerical values directly as additional text in the model input [18].

We also ran evidence-context ablations. In reduced-evidence settings, we used fewer evidence snippets and shorter evidence character limits. In long-context settings, we increased the amount of evidence passed to the model. These experiments tested how much the ranking model depended on evidence coverage.

# 5. Experimental Setup

## 5.1. Data Files

For the submitted trace scorer, we used the combined multilingual training data containing English, Spanish, and Arabic claims. For the DPO experiments, we used an expanded English dataset, constructed by combining the original English examples with Google Translate translations of the Arabic and Spanish examples into English. We also ran translated prompt-baseline invariance checks on Arabic and Spanish validation examples.

## 5.2. Compute Resources

All experiments were run on the SCC GPU cluster using NVIDIA A100 GPUs. The main trace-scorer training runs used distributed training with `torchrun` on two GPUs, while inference and smaller

ablation runs were executed on a single GPU when possible. We used Python 3.11 with uv for environment management, PyTorch and Hugging Face Transformers for model training, and Weights & Biases for experiment tracking. To reduce memory usage, the LLM-based trace scorers were trained with 4-bit loading and QLoRA adapters, and CUDA expandable memory segments were enabled during training. Overall, these experiments consumed approximately 42,520 core-hours on the SCC GPU cluster, reflecting the computational cost of comparing prompt-based ranking, DPO fine-tuning, and supervised trace-scoring variants.

### 5.3. Models and Training

The submitted system used `Llama-3.1-8B-Instruct` with QLoRA, 4-bit loading, LoRA adapters over all linear modules, and `sdpa` attention. The main trace scorer used a two-logit cross-entropy head. We trained for two epochs with learning rate $1e^{-4}$, warmup ratio 0.03, per-device batch size 8, gradient accumulation 1, and early stopping with patience 2. The best checkpoint was selected using positive F1.

For additional trace-scorer variants, we tested a one-logit binary cross-entropy head, different LoRA target modules, LoRA dropout, weight decay, cosine scheduling, numeric embeddings, normalized number hints, and reduced-evidence settings. The DPO experiments used `Mistral-7B-Instruct-v0.3` initialized from an English SFT LoRA adapter.

### 5.4. Input Settings

For the submitted long-context trace scorer, we included up to 20 evidence snippets per claim, truncated each evidence snippet to 3078 characters, and set the maximum tokenized input length to 15000 tokens. This was intended to preserve as much of the constructed claim-evidence-trace input as possible. Reduced-evidence ablations used fewer evidence snippets and shorter character limits.

### 5.5. Inference and Evaluation

At inference time, the trace scorer scored each candidate trace independently. Traces were ranked by positive-class score, and the final claim verdict was derived from the top five trace verdicts. Prompted and DPO models instead generated JSON rankings and verdicts directly. We report Recall@5, macro-F1, MRR@5, and the average score, where the average score corresponds to the mean of macro-F1 and Recall@5, following the official language-specific evaluation.

## 6. Results and Discussion

On the official leaderboard, our submitted trace-scoring system achieved its strongest relative performance on English, where it ranked first. It also ranked third on Spanish and sixth on Arabic. Table 1 reports the corresponding official leaderboard snapshot.

We report validation results in Tables 2, 4, and 6, and official test-set results in Tables 3, 5, and 7. The validation tables document model selection and ablation behavior, while the test tables provide a more generalizable assessment and allow comparison with the official task results. In each table, underlining marks the model selected for the final submission, while boldface marks the best result for that language and split. In the tables, Mean F1/R@5 denotes the task average score, computed separately for each language as $(\text{Macro-F1} + \text{Recall@5})/2$.

The results indicate that supervised trace scoring was the most effective formulation among the approaches we explored. The zero-shot prompted baselines were simple to apply, but the listwise setting showed strong positional bias: the generated ranking was often influenced by the order of candidate traces in the prompt. Pairwise prompting reduced the size of each decision by comparing two traces at a time, and the balanced and bidirectional variants were designed to reduce A/B ordering effects. However, positional bias was still present, and the approach required many model calls for each claim.

**Table 1**
Leaderboard snapshot for the submitted GippLab system.

| Language | Mean F1/R@5 | Rank |
|---|---|---|
| English | 0.4286 | 1 |
| Spanish | 0.4777 | 3 |
| Arabic | 0.5873 | 6 |

**Table 2**
English validation results.

| Model | Recall@5 | Macro-F1 | Mean F1/R@5 | MRR@5 |
|---|---|---|---|---|
| Random baseline | 0.2326 | 0.4673 | 0.3500 | 0.5451 |
| Prompt baseline | 0.2215 | 0.4924 | 0.3569 | 0.6197 |
| English trace scorer | 0.2969 | 0.4967 | 0.3968 | 0.5900 |
| English trace scorer (Numeric embedding) | 0.2437 | 0.5579 | 0.3550 | 0.5579 |
| English DPO | 0.2345 | 0.4709 | 0.3528 | 0.5493 |
| Multilingual trace scorer (Mistral-7B) | 0.2996 | 0.5025 | 0.4010 | 0.5947 |
| Multilingual trace scorer (Llama-3.1-8B) | 0.3180 | 0.5002 | 0.4091 | 0.5985 |
| Multilingual trace scorer (all evidence, Llama-3.1-8B) | 0.3103 | 0.5195 | 0.4163 | 0.5943 |
| Multilingual trace scorer (regularisation, Llama-3.1-8B) | 0.3184 | 0.5143 | 0.4163 | 0.6032 |
| **Multilingual trace scorer (regularisation + num norm, Llama-3.1-8B)** | **0.3232** | **0.5168** | **0.4200** | **0.6094** |
| Multilingual trace scorer (regularisation + BCE, Llama-3.1-8B) | 0.3182 | 0.5200 | 0.4192 | 0.6057 |

**Table 3**
English test results.

| Model | Recall@5 | Macro-F1 | Mean F1/R@5 | MRR@5 |
|---|---|---|---|---|
| Random baseline | 0.2161 | 0.5914 | 0.4037 | 0.7936 |
| Prompt baseline | 0.2204 | 0.5864 | 0.4034 | 0.7979 |
| English trace scorer | 0.2163 | 0.5885 | 0.4024 | 0.7880 |
| English trace scorer (Numeric embedding) | 0.2187 | 0.5922 | 0.4054 | 0.7949 |
| English DPO | 0.2175 | 0.6015 | 0.4095 | 0.7909 |
| Multilingual trace scorer (Mistral-7B) | 0.2130 | 0.5861 | 0.3995 | 0.7819 |
| Multilingual trace scorer (Llama-3.1-8B) | 0.2376 | 0.5881 | 0.4129 | 0.7878 |
| **Multilingual trace scorer (all evidence, Llama-3.1-8B)** | **0.2421** | **0.6150** | **0.4286** | **0.7849** |
| Multilingual trace scorer (regularisation, Llama-3.1-8B) | 0.2154 | 0.5880 | 0.4017 | 0.7865 |
| Multilingual trace scorer (regularisation + num norm, Llama-3.1-8B) | 0.2425 | 0.5997 | 0.4211 | 0.8007 |
| Multilingual trace scorer (regularisation + BCE, Llama-3.1-8B) | 0.2515 | 0.5897 | 0.4206 | 0.8133 |

The validation Recall@5 scores were structurally limited by the available candidate traces and the fixed top-5 cutoff. Even with a perfect ranking, the maximum achievable Recall@5 was 0.39 for English, 0.32 for Spanish, and 0.28 for Arabic. Assuming perfect claim-verdict classification, these correspond to theoretical maximum average scores of 0.695, 0.660, and 0.640, respectively. This indicates that the final score was limited not only by model performance, but also by how many gold-matching traces were available for each claim, with stronger limitations for Spanish and Arabic.

The DPO model addressed ranking more directly by training on preferred and rejected ranking

**Table 4**
Spanish validation results.

| Model | Recall@5 | Macro-F1 | Mean F1/R@5 | MRR@5 |
|---|---|---|---|---|
| Random baseline | 0.2019 | 0.3535 | 0.2778 | 0.6761 |
| Prompt baseline | 0.1713 | 0.2500 | 0.2107 | 0.6200 |
| Spanish trace scorer | 0.2695 | 0.3613 | 0.3154 | 0.7406 |
| Spanish trace scorer (Numeric embedding) | 0.2688 | 0.3647 | 0.3168 | 0.7442 |
| English DPO on Spanish | 0.2018 | 0.3560 | 0.2789 | 0.6697 |
| Multilingual trace scorer (Mistral-7B) | 0.2697 | 0.3729 | 0.3213 | 0.7423 |
| Multilingual trace scorer (Llama-3.1-8B) | 0.2754 | 0.3919 | 0.3337 | 0.7464 |
| Multilingual trace scorer (all evidence, Mistral-7B) | 0.2822 | 0.4008 | 0.3415 | 0.7481 |
| **Multilingual trace scorer (all evidence, Llama-3.1-8B)** | **0.2787** | **0.4284** | **0.3553** | **0.7429** |
| Multilingual trace scorer (regularisation, Llama-3.1-8B) | 0.2745 | 0.3864 | 0.3304 | 0.7398 |
| Multilingual trace scorer (regularisation + num norm, Llama-3.1-8B) | 0.2662 | 0.3920 | 0.3291 | 0.7275 |
| Multilingual trace scorer (regularisation + BCE, Llama-3.1-8B) | 0.2680 | 0.4088 | 0.3384 | 0.7335 |

**Table 5**
Spanish test results.

| Model | Recall@5 | Macro-F1 | Mean F1/R@5 | MRR@5 |
|---|---|---|---|---|
| Random baseline | 0.2184 | 0.5683 | 0.3934 | 0.7567 |
| Prompt baseline | 0.2221 | 0.5578 | 0.3889 | 0.7418 |
| Spanish trace scorer | 0.2338 | 0.5758 | 0.4048 | 0.7774 |
| Spanish trace scorer (Numeric embedding) | 0.2102 | 0.5501 | 0.3801 | 0.7467 |
| English DPO on Spanish | 0.2290 | 0.5741 | 0.4016 | 0.7650 |
| Multilingual trace scorer (Mistral-7B) | 0.2141 | 0.5655 | 0.3898 | 0.7586 |
| Multilingual trace scorer (Llama-3.1-8B) | 0.2679 | 0.5770 | 0.4224 | 0.7857 |
| **Multilingual trace scorer (all evidence, Llama-3.1-8B)** | **0.3046** | **0.6508** | **0.4777** | **0.8409** |
| Multilingual trace scorer (regularisation, Llama-3.1-8B) | 0.2140 | 0.5584 | 0.3862 | 0.7494 |
| Multilingual trace scorer (regularisation + num norm, Llama-3.1-8B) | 0.2936 | 0.6389 | 0.4662 | 0.8201 |
| Multilingual trace scorer (regularisation + BCE, Llama-3.1-8B) | 0.2958 | 0.6326 | 0.4642 | 0.8283 |

outputs. It used hard negative outputs, such as rankings that placed an incorrect trace first or predicted an incorrect verdict. However, DPO still relied on a generative ranking objective: the model had to produce a complete JSON ranking and verdict. This remained sensitive to prompt formatting and did not directly optimize independent trace-level relevance scores or the official Recall@5 metric.

The final trace-scoring system avoided these issues by scoring each candidate trace independently. This produced a direct positive-class score for every trace, allowing the trace ranking to be constructed deterministically. The same framework also made it straightforward to use long evidence contexts and multilingual training. The test results show especially strong generalization for Spanish and Arabic, where the all-evidence Llama model substantially outperformed the baselines and language-specific trace scorers.

One dataset-level challenge was that some examples had a gold claim label that did not match any of the provided trace verdicts. This issue was present across all splits, but it was more frequent in training and validation than in the official test set. In the training data, 30.0% of English examples, 19.4% of

**Table 6**
Arabic validation results.

| Model | Recall@5 | Macro-F1 | Mean F1/R@5 | MRR@5 |
|---|---|---|---|---|
| Random baseline | 0.2017 | 0.7203 | 0.4719 | 0.7203 |
| Prompt baseline | 0.2513 | 0.7386 | 0.4950 | 0.8200 |
| Arabic trace scorer | 0.2526 | 0.7482 | 0.5004 | 0.7731 |
| Arabic trace scorer (Numeric embedding) | 0.2095 | 0.7454 | 0.4774 | 0.7257 |
| English DPO on Arabic | 0.2095 | 0.7454 | 0.4774 | 0.7257 |
| Multilingual trace scorer (Mistral-7B) | 0.2520 | 0.7511 | 0.5016 | 0.7746 |
| Multilingual trace scorer (Llama-3.1-8B) | 0.2739 | 0.7645 | 0.5192 | 0.8010 |
| Multilingual trace scorer (all evidence, Mistral-7B) | 0.2726 | 0.7816 | 0.5271 | 0.8006 |
| Multilingual trace scorer (all evidence, Llama-3.1-8B) | 0.2765 | 0.7905 | 0.5333 | 0.8037 |
| **Multilingual trace scorer (regularisation, Llama-3.1-8B)** | **0.2793** | **0.7960** | **0.5377** | **0.8087** |
| Multilingual trace scorer (regularisation + num norm, Llama-3.1-8B) | 0.2759 | 0.7804 | 0.5281 | 0.8048 |
| Multilingual trace scorer (regularisation + BCE, Llama-3.1-8B) | 0.2788 | 0.7943 | 0.5365 | 0.8083 |

**Table 7**
Arabic test results.

| Model | Recall@5 | Macro-F1 | Mean F1/R@5 | MRR@5 |
|---|---|---|---|---|
| Random baseline | 0.2150 | 0.7099 | 0.4625 | 0.7181 |
| Prompt baseline | 0.1945 | 0.6672 | 0.4309 | 0.6903 |
| Arabic trace scorer | 0.2200 | 0.7288 | 0.4744 | 0.7384 |
| Arabic trace scorer (Numeric embedding) | 0.2354 | 0.7352 | 0.4853 | 0.7370 |
| English DPO on Arabic | 0.2354 | 0.7352 | 0.4853 | 0.7370 |
| Multilingual trace scorer (Mistral-7B) | 0.2354 | 0.7514 | 0.4934 | 0.7515 |
| Multilingual trace scorer (Llama-3.1-8B) | 0.3303 | 0.8036 | 0.5669 | 0.8470 |
| **Multilingual trace scorer (all evidence, Llama-3.1-8B)** | **0.3406** | **0.8340** | **0.5873** | **0.8611** |
| Multilingual trace scorer (regularisation, Llama-3.1-8B) | 0.2173 | 0.6953 | 0.4563 | 0.7209 |
| Multilingual trace scorer (regularisation + num norm, Llama-3.1-8B) | 0.3406 | 0.7994 | 0.5700 | 0.8611 |
| Multilingual trace scorer (regularisation + BCE, Llama-3.1-8B) | 0.3406 | 0.8006 | 0.5706 | 0.8611 |

Spanish examples, and 19.2% of Arabic examples had no trace verdict matching the gold claim label. In validation, the corresponding rates were 29.4% for English, 19.6% for Spanish, and 18.9% for Arabic. On the test set, the mismatch rate was lower and more balanced across languages: 13.9% for English, 13.1% for Spanish, and 13.7% for Arabic. These cases are difficult for supervised trace scoring because no candidate trace can be labeled as positive. Improving consistency between claim-level labels and trace-level verdicts would likely make trace-ranking approaches more reliable.

Across the validation experiments, `Llama-3.1-8B-Instruct` generally performed better than the earlier Mistral-based variants. The all-evidence setting was also important: instead of using only a small number of evidence snippets, we provided up to all evidence items and used a long maximum sequence length so that most of the constructed input could pass through the model. This improved the model's ability to compare the claim against a broader evidence context.

We also tested regularized variants of the Llama trace scorer. In these runs, we used LoRA dropout of 0.10 and weight decay of 0.01. We additionally evaluated a binary cross-entropy version of the

trace scorer, in contrast to the default two-logit cross-entropy formulation. These variants improved validation performance in several settings, especially on Arabic, where the regularized Llama models achieved the strongest validation scores. On the test set, however, the all-evidence Llama model was the strongest overall system, while the regularized and normalized-number variants remained competitive.

The weaker relative Arabic leaderboard rank likely reflects a combination of data, modeling, and preprocessing factors rather than a single cause. Arabic differs more substantially from English and Spanish in script, morphology, and tokenization behavior, which can make evidence matching and numerical expression comparison harder for a shared multilingual model. The dataset statistics also show several relevant distributional differences: the Arabic files contain only true and false claim labels, whereas English and Spanish also contain conflicting examples, making macro-F1 less directly comparable across languages; Arabic had the lowest validation Recall@5 upper bound (0.28), indicating that fewer gold-matching traces could be recovered within the top five even under a perfect ranking; and Arabic test examples contained much shorter evidence and trace text by character count, with an average total evidence length of 692 characters compared with 10,042 for English and 10,819 for Spanish, and an average gold-matching trace length of 308 characters compared with 679 for English and 561 for Spanish. Together, these factors help explain why the same all-evidence Llama trace scorer generalized well overall but produced a weaker relative leaderboard rank on Arabic.

Based on the average score across the three validation languages, we selected the multilingual trace scorer with all evidence based on `Llama-3.1-8B` as the final submitted model, rather than selecting a separate best model for each language. This was a conservative choice because language-specific selection on the validation set could overfit to that split. The official test results support this choice: the all-evidence Llama scorer achieved the best average score on English, Spanish and Arabic test data.

## 7. Conclusion

We presented the GippLab system for CheckThat! 2026 Task 2 on multilingual numerical claim verification. We explored several ranking-based approaches, including zero-shot listwise prompting, pairwise prompting, DPO ranking, and supervised trace scoring. Our final approach therefore used supervised trace scoring, where each candidate reasoning trace was scored independently and the final verdict was derived from the top-ranked traces.

The submitted system fine-tuned `Llama-3.1-8B-Instruct` with QLoRA on combined English, Spanish, and Arabic data. This formulation provided the strongest overall results among our experiments and ranked first on English, third on Spanish, and sixth on Arabic on the official leaderboard. Our experiments also showed that long-context evidence, multilingual training, and regularized Llama-based trace scorers were important for strong validation performance.

Future work could improve the approach by directly optimizing ranking metrics such as Recall@5, handling examples where no trace verdict matches the gold claim label, and developing more robust methods for handling numerical expressions across reasoning traces.

## Acknowledgments

We thank the CheckThat! 2026 Task 2 organizers for providing the dataset, evaluation setup, and leaderboard infrastructure. We also thank the GippLab group for their feedback and support during the development of this system. We acknowledge the computing resources provided by the University of Göttingen.

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to assist with language formatting and grammar correction. The authors reviewed and edited the generated content and take full responsibility for the final publication.

## References


[1] R. Rahman, A. A. Mishra, A fragile number sense: Probing the elemental limits of numerical reasoning in llms, 2025. URL: https://arxiv.org/abs/2509.06332. arXiv:2509.06332.

[2] Y. Wang, P. Zhang, J. Tang, H.-R. Wei, B. Yang, R. Wang, C. Sun, F. Sun, J. Zhang, J. Wu, Q. Cang, Y. Zhang, J. Lin, F. Huang, J. Zhou, Polymath: Evaluating mathematical reasoning in multilingual contexts, in: D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, N. Chen (Eds.), Advances in Neural Information Processing Systems, volume 38, Curran Associates, Inc., 2025. URL: https://proceedings.neurips.cc/paper_files/paper/2025/file/3d5aa9a7ce28cdc710fbd044fd3610f3-Paper-Datasets_and_Benchmarks_Track.pdf.

[3] 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.

[4] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, et al., Large language models are effective text rankers with pairwise ranking prompting, in: Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 1504–1518.

[5] L. Shi, C. Ma, W. Liang, X. Diao, W. Ma, S. Vosoughi, Judging the judges: A systematic study of position bias in llm-as-a-judge, 2025. URL: https://arxiv.org/abs/2406.07791. arXiv:2406.07791.

[6] J. Dang, A. Ahmadian, K. Marchisio, J. Kreutzer, A. Üstün, S. Hooker, Rlhf can speak many languages: Unlocking multilingual preference optimization for llms, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 13134–13156.

[7] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, 2023. URL: https://arxiv.org/abs/2305.14314. arXiv:2305.14314.

[8] Meta AI, Meta llama 3.1 8b instruct, https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, 2024. Accessed: 2026-05-17.

[9] P. Chungkham, V. Viswanathan, V. Setty, A. Anand, Think right, not more: Test-time scaling for numerical claim verification, in: Empirical Methods in Natural Language Processing (EMNLP) 2025, Association for Computational Linguistics, 2025, pp. 24345–24363.

[10] V. V, A. Anand, A. Anand, V. Setty, Quantemp: A real-world open-domain benchmark for fact-checking numerical claims, 2024. URL: https://arxiv.org/abs/2403.17169. arXiv:2403.17169.

[11] T. Liu, Z. Qin, J. Wu, J. Shen, M. Khalman, R. Joshi, Y. Zhao, M. Saleh, S. Baumgartner, J. Liu, P. J. Liu, X. Wang, LiPO: Listwise preference optimization through learning-to-rank, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 2404–2420. URL: https://aclanthology.org/2025.naacl-long.121/. doi:10.18653/v1/2025.naacl-long.121.

[12] J. Qiao, J. Huang, X. Ma, S. Wang, D. Yin, E. Kanoulas, A. Yates, Llm-based listwise reranking under the effect of positional bias, 2026. URL: https://arxiv.org/abs/2604.03642. arXiv:2604.03642.

[13] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, C. Finn, Direct preference optimization: Your language model is secretly a reward model, 2024. URL: https://arxiv.org/abs/2305.18290. arXiv:2305.18290.

[14] 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, 2021. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[15] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[16] PyTorch Contributors, torch.nn.functional.scaled_dot_product_attention, https://pytorch.org/docs/stable/generated/torch.nn.functional.scaled_dot_product_attention.html, 2024. Accessed: 2026-05-21.

[17] A. Dutulescu, S. Ruseti, M. Dascalu, Value-aware numerical representations for transformer language models, arXiv preprint arXiv:2601.09706 (2026).

[18] E. Schwartz, L. Choshen, J. Shtok, S. Doveh, L. Karlinsky, A. Arbelle, Numerologic: Number encoding for enhanced llms' numerical reasoning, 2024. URL: https://arxiv.org/abs/2404.00459. arXiv:2404.00459.