nadie at CheckThat! 2026: Numerical Reasoning-Enhanced Verifier for Fact-Checking Numerical Claims

Yu Miao[1,*], Maxim Emelianov[2]

[1]Department of Computational Linguistics, University of Zurich, Zurich, Switzerland
[2]Department of Informatics, University of Zurich, Zurich, Switzerland

Abstract

This paper describes the system submitted by team *nadie* to Task 2, Fact-Checking Numerical Claims, of the CheckThat! 2026 Lab at CLEF. The task requires ranking multiple LLM-generated reasoning traces for a numerical claim and deriving a final veracity label. Our final system uses a DeBERTa-v3-large pointwise verifier with Numerical Chain-of-Thought (NumCoT) input augmentation, class-weighted binary cross-entropy training, and top-5 softmax-weighted verdict aggregation. NumCoT extracts numerical expressions from the claim and candidate justification and inserts explicit compatible pair comparisons into the verifier input. We also compare pointwise, pairwise, listwise, evidence-aware, and ensemble variants. The final system achieves an official leaderboard combined score of **0.422**, compared with **0.408** for the organiser baseline. Our results indicate that increasing encoder capacity produces the largest improvement, whereas NumCoT provides a smaller, low-cost gain. Evidence-aware and ensemble systems are competitive, but do not outperform the final single-model verifier.

Keywords

fact verification, numerical reasoning, reasoning-trace ranking, DeBERTa, test-time scaling, CheckThat! 2026

1. Introduction

Automated fact verification is commonly formulated as the joint problem of finding relevant evidence and predicting whether that evidence supports or refutes a claim [1]. Numerical claims make this setting especially difficult. Small changes in quantities, units, denominators, or time periods can reverse a verdict while leaving most of the wording unchanged. The QuanTemp benchmark consequently focuses on naturally occurring claims containing quantities and temporal expressions and shows that standard fact-verification systems remain limited in this setting [2].

Large language models can produce natural-language rationales that decompose a verification decision, but fluent rationales are not necessarily correct. Generating several candidate traces at inference time creates another problem: a verifier must distinguish useful reasoning from plausible but incorrect reasoning. CheckThat! 2026 Task 2 formalises this test-time scaling setting by providing a claim, associated evidence, and multiple generated reasoning traces. Systems must rank the traces and infer one final verdict from the highest-ranked candidates [3].

The organiser baseline independently scores each trace with a binary cross-entropy (BCE) verifier and adopts the verdict of the highest-scoring trace. Our system investigates whether explicitly exposing numerical comparisons, changing the ranking objective, incorporating retrieved evidence, and aggregating several top-ranked verdicts improve this baseline.

Our contributions are as follows:

- Introducing NumCoT, a lightweight input augmentation that identifies numerical expressions and verbalises compatible claim–justification comparisons for an encoder-based verifier.
- Comparing pointwise BCE, pairwise margin, listwise softmax, and evidence-aware verification under the same shared-task setting.
- Replacing top-1 verdict selection with top-5 score-weighted voting and analyse the effects of model scale, trace coverage, evidence, and ensembling.

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany
*Corresponding author.
miao.yu2@uzh.ch (Y. Miao); maxim.emelianov@uzh.ch (M. Emelianov)

- The DeBERTa-v3-large system improves the official combined score from 0.408 for the organiser baseline to 0.422.

2. Related Work

Automated fact verification. FEVER established a widely used benchmark combining evidence retrieval with three-way claim verification [1]. Later benchmarks broadened the evidence types and reasoning demands; FEVEROUS, for example, includes both structured and unstructured evidence [4]. In contrast to synthetic Wikipedia claims, QuanTemp contains real-world numerical and temporal claims, fine-grained metadata, and web evidence collected without claim-label leakage [2]. The present shared task reuses the English QuanTemp claims while changing the target from direct claim classification to joint reasoning-trace ranking and verdict prediction.

Numerical reasoning. Numerical language understanding requires operations that are difficult to recover from lexical overlap alone. DROP evaluates discrete reasoning over paragraphs, including addition, counting, and comparison [5], while NumNet augments reading comprehension with an explicit numerical graph to model relations among quantities [6]. Our NumCoT method is more lightweight: it does not alter the encoder architecture, but appends deterministically extracted numerical comparisons to the input.

Reasoning-trace verification. Sampling several reasoning paths and selecting among them is a common way to spend additional inference-time computation. Self-consistency aggregates answers over diverse sampled chains [7], whereas verifier-based approaches learn to score candidate solutions or individual reasoning steps [8, 9]. Our setting differs because the candidate traces include categorical fact-checking verdicts and the evaluation jointly rewards retrieval of relevant traces and the final three-way verdict.

3. Task and Data

For a claim c, the input contains a set of generated traces $T_c = \{t_1, \ldots, t_n\}$, associated evidence documents, and candidate verdicts $v_i \in \{\text{True}, \text{False}, \text{Conflicting}\}$. A training trace is defined as relevant when its candidate verdict equals the claim's gold label. The system returns a score for every trace and one final verdict.

Trace ranking is evaluated with Recall@5 and verdict classification with Macro-F1. The official combined measure is

$$\text{Combined} = \frac{\text{Recall@5} + \text{MacroF1}}{2}. \tag{1}$$

Equation 1 gives equal weight to ranking and classification.

We focus on English. The supplied training file contains 6,400 claims and 127,020 traces; each claim has 15–20 traces (19.85 on average). Candidate verdicts match the gold label for 47.83% of traces and disagree for 52.17%. Thus, more than half of the available traces are negative under the task's relevance definition. Table 1 summarises these statistics.

The central challenge is that candidate traces are often linguistically fluent even when their verdict is incorrect. Errors include reversed comparisons, incompatible units, mismatched temporal scope, unsupported inferences, and disagreement between the justification and its stated verdict.

4. Method

Figure 1 presents the final inference pipeline. Each trace is augmented and scored independently; trace scores determine both the ranking and the weighted verdict vote.

Table 1
Statistics of the English training data used in our experiments.

Property	Value
Claims	6,400
Reasoning traces	127,020
Traces per claim	15–20
Mean traces per claim	19.85
Gold-matching verdicts	47.83%
Non-matching verdicts	52.17%

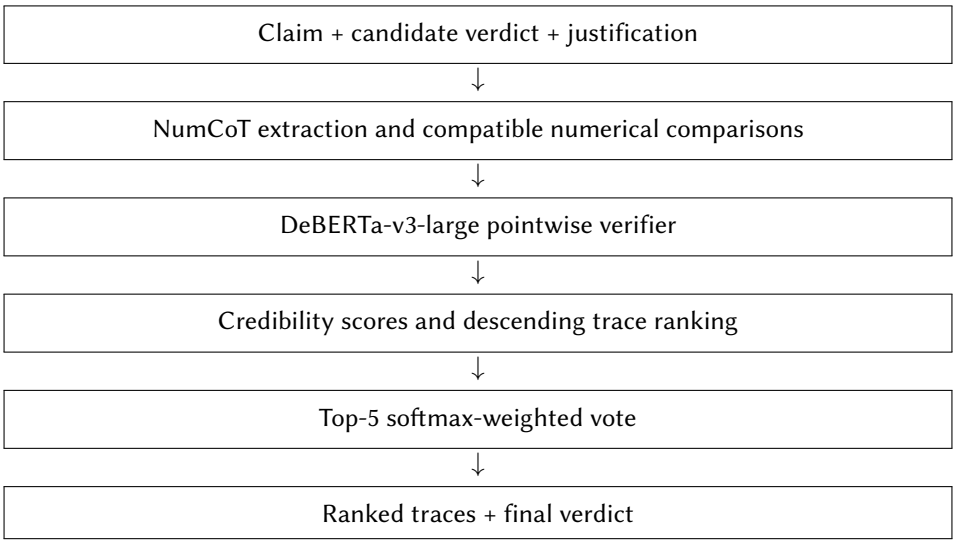

Figure 1: Final system pipeline. Evidence is available in the task data but is not directly encoded by the final submitted model.

4.1. Pointwise Verifier

For every trace, the final model predicts whether its candidate verdict matches the gold label. Its input contains the claim, candidate verdict, justification, and the NumCoT block described in Section 4.2. We use DeBERTa-v3-large [10]. The first-token representation $\mathbf{h}_i$ is passed through a two-layer scoring head:

$$z_i = W_2 \,\mathrm{Dropout}(\mathrm{GELU}(W_1 \mathbf{h}_i + b_1)) + b_2. \tag{2}$$

The hidden layer has 256 units and dropout probability 0.1. At inference time, the credibility score is $s_i = \sigma(z_i)$.

We optimise class-weighted binary cross-entropy:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \left[w_+ y_i \log s_i + (1 - y_i) \log(1 - s_i) \right], \tag{3}$$

where $y_i = 1$ when the candidate verdict matches the gold label and $w_+ = n_{\mathrm{neg}}/n_{\mathrm{pos}}$ compensates for class imbalance.

We sample at most six traces per claim. If at least two positives are available, we sample two positives and four negatives; with one positive, we sample that trace and five negatives; without positives, we sample six negatives. This strategy avoids allowing claims with 20 traces to dominate a batch while retaining several contrasting verdicts. Train/validation splitting is performed at the claim level so that traces from one claim cannot occur in both partitions.

4.2. Numerical Chain-of-Thought

NumCoT is a deterministic input transformation rather than a generated rationale. It first extracts percentages, four-digit years, decimal values, and integers from the claim and justification. It then filters incompatible types, excludes year–year comparisons, and removes extremely mismatched pairs. For at most five retained pairs, it appends an explicit relation such as *"45% (45.0) > 32% (32.0), difference = 13"*. The resulting input is:

```
Claim: <claim>
Verdict: <verdict>
Justification: <justification>
[Numerical Reasoning Steps]
Claim nums: ...   Justification nums: ...
Comparisons: ...
```

Because the procedure uses regular expressions and arithmetic comparisons, it adds little runtime cost and does not require changes to the encoder.

4.3. Alternative Ranking Objectives

Pairwise margin ranking. For each claim, we form positive–negative trace pairs and retain at most 20 pairs. Their order is randomised. A shared DeBERTa-v3-base scorer produces z_a and z_b, and the model is trained with margin $m = 0.3$:

$$\mathcal{L}_{\text{pair}} = \max\left(0, m - r_{ab}(z_a - z_b)\right), \tag{4}$$

where $r_{ab} = 1$ if trace a is relevant and trace b is not, and -1 for the reversed ordering. This directly learns within-claim preferences, whereas the pointwise objective learns an absolute relevance probability.

Listwise softmax ranking. The listwise variant retains all traces for each claim and jointly normalises their logits. With binary relevance y_{ci} for trace i of claim c, our implemented listwise objective is

$$\mathcal{L}_{\text{list}} = -\frac{1}{|C|} \sum_{c \in C} \sum_{i=1}^{|T_c|} y_{ci} \log \frac{\exp(z_{ci})}{\sum_j \exp(z_{cj})}. \tag{5}$$

This objective distributes probability mass over the relevant members of a claim-level list. It is a relevance-weighted listwise softmax loss rather than the permutation-based ListMLE objective.

4.4. Evidence-Aware and Ensemble Variants

The Longformer variant concatenates the claim, retrieved evidence, candidate verdict, justification, and NumCoT and encodes up to 1,024 tokens with Longformer-base [11]. It uses the same class-weighted BCE objective as Equation 3 and the same six-trace sampling policy. An additional all-traces evidence model uses DeBERTa-v3-base, keeps every available trace as a separate instance, and limits evidence to the first three strings truncated to 700 characters each. These experiments test whether broader trace coverage and direct evidence access compensate for the extra input noise.

Finally, we construct a four-model ensemble from the NumCoT, pairwise, Longformer, and all-traces evidence systems. Scores are aligned by exact trace text, normalised within each claim, and linearly combined. Model weights, aggregation strategy, and softmax temperature are selected on validation data by grid search. The selected weights are 0.33, 0.27, 0.10, and 0.30, respectively, with top-5 voting and temperature 0.5.

Table 2
Official leaderboard combined scores. The combined score averages Macro-F1 and Recall@5; higher is better.

System	Combined
Organiser baseline (BCE + top-1)	0.408
NumCoT + BCE-base	0.410
Pairwise margin ranking	0.413
Pairwise + top-5 voting	0.414
Longformer + evidence	0.411
Listwise softmax ranking	0.408
All-traces evidence model	0.421
Calibrated four-model ensemble	0.422
Final: BCE-large + NumCoT + softmax voting	**0.422**

4.5. Verdict Aggregation

The final system ranks traces by descending s_i. For classification, it selects the top K traces and computes

$$w_i = \frac{\exp(s_i/\tau)}{\sum_{j \in \text{top}K} \exp(s_j/\tau)}, \tag{6}$$

followed by

$$S(v) = \sum_{i \in \text{top}K} \mathbb{I}[v_i = v] w_i, \qquad \hat{v} = \arg\max_v S(v). \tag{7}$$

Here w_i is the normalised trace weight and $S(v)$ is the total support for verdict v. The final submission uses $K = 5$ and $\tau = 1.0$. This aggregation reduces dependence on one potentially noisy top-ranked trace.

5. Experimental Setup

All experiments use the English sub-task. The final model has maximum input length 512, learning rate 5×10^{-6}, batch size 4, four gradient accumulation steps (effective batch size 16), and four epochs. We use AdamW with weight decay 0.01, cosine scheduling, 10% warmup, gradient clipping at 1.0, and gradient checkpointing. The best checkpoint is selected on the claim-level validation split. Inference applies a sigmoid to each trace logit before ranking and voting.

The final DeBERTa-v3-large system does not directly encode the supplied evidence. Evidence is tested only in the Longformer and all-traces evidence variants. This distinction is important: the final verifier evaluates the internal quality and verdict of a generated trace, whereas the evidence-aware variants can additionally compare that trace with retrieved material.

6. Results

Table 2 reports official leaderboard combined scores rounded to three decimal places. Only the combined measure was exposed for these submissions, so separate official Macro-F1 and Recall@5 values are unavailable. Validation component metrics are therefore not mixed with leaderboard results.

The final model and ensemble both round to 0.422, compared with 0.408 for the baseline. Thus, the final model improves over the baseline by approximately 0.014 absolute. The underlying scores of the two strongest systems differ only marginally, so we do not interpret their ordering as evidence of a meaningful difference; the single-model system was selected because it is simpler to train and deploy.

7. Analysis

The largest single improvement in our experiments comes from encoder scale: the base NumCoT verifier reaches 0.410, while the large verifier reaches 0.422, a difference of 0.012 that is substantially larger than any other gain we observe (though the two models were trained separately, so this is not a controlled ablation). Candidate traces often fail for subtle reasons – a comparison across incompatible denominators, a claim placed in the wrong time period, or a verdict that outruns what the justification actually supports – and a larger encoder seems better placed to catch this kind of error than a smaller one.

Isolating the effect of NumCoT itself is harder than it first appears. Our own NumCoT+BCE-base system scores 0.410, only 0.002 above the organiser baseline's 0.408, but the baseline notebook specifies Llama-3.2-1B, a decoder-only model, as its backbone, while our comparison system uses a DeBERTa-v3-base encoder. We did not replicate the baseline's exact backbone because Llama-3.2-1B is gated and we did not have access to it. This 0.002 gap therefore reflects a difference in architecture and scale at least as much as it reflects NumCoT, and should not be read as an ablation of NumCoT in isolation. What NumCoT does appear to do, based on its explicit comparisons, is make magnitude relations easier for the encoder to notice; what it cannot do is resolve unit mismatches, changes in temporal scope, or whether the cited evidence is sufficient in the first place. We treat it as a cheap, useful signal, not a substitute for genuine numerical reasoning. We also acknowledge that we cannot cleanly quantify its contribution from the comparisons available to us.

Pairwise training improves over the pointwise NumCoT model, particularly once combined with top-5 aggregation, but our listwise variant does not. Equation 5 penalises every relevant trace independently rather than normalising by the number of positives per claim, so claims with many relevant traces contribute disproportionately to the loss; this objective is also a relevance-weighted softmax rather than the permutation-based ListMLE loss, so the comparison is specific to our implementation rather than a verdict on listwise ranking in general.

Qualitative NumCoT and trace-noise analysis. The same inspection turned up cases where NumCoT can also add noise. In a surgical-mask trial example, the string "6,000" is correctly parsed as 6000, but the extractor also pulls out meaningless sub-spans such as "6" and "000", producing irrelevant comparisons between 6 and 000 that have nothing to do with the claim. In a separate mask-pathogen example, traces latch onto surface agreement – six masks, at least 11 pathogens – and predict True, when the gold label is Conflicting. What both cases share is that the traces settle on a confident True/False verdict for claims that should really be flagged as only partially supported, which is exactly the situation top-5 weighted voting is meant to guard against. We stop short of claiming a controlled trend here: a systematic sweep over 5, 10, 15, and all traces per claim would be needed to show how Recall@5 and Macro-F1 degrade with trace count or noise, and we leave that for future work. Evidence only pays off once it is kept under control. Feeding the Longformer variant the full evidence text yields 0.411, only 0.003 above the baseline – which suggests that a longer context window mostly adds irrelevant material. The all-traces model, which truncates and restricts evidence more aggressively, reaches 0.421 instead, matching the pattern that evidence selection matters more than raw context length. That model also changes trace coverage at the same time, though, so this comparison does not cleanly separate the effect of evidence truncation from the effect of using more traces.

Finally, aggregation matters because the traces themselves are noisy: only 47.83% carry the gold verdict, so a top-1 decision can be thrown off by a single incorrectly promoted trace. Top-5 voting lifts the pairwise model from 0.413 to 0.414, and we keep it in the final system for that reason – though since the leaderboard only exposes the combined score, we cannot say whether this gain comes from Recall@5, Macro-F1, or both.

8. Limitations and Future Work

Several of our comparisons are not fully controlled. The organiser baseline notebook specifies a Llama-3.2-1B decoder as its backbone, whereas our NumCoT+BCE-base system uses a DeBERTa-v3-base encoder; we did not replicate the organiser's exact backbone because Llama-3.2-1B is a gated model to which we did not have access. The 0.002 gap between the two systems therefore reflects a difference in architecture and scale at least as much as it reflects the effect of NumCoT itself, and should not be read as an ablation of NumCoT in isolation. The all-traces evidence model changes both trace coverage and evidence context at once, and the final large model changes encoder capacity in addition to adding NumCoT, so neither isolates a single factor either. Our official leaderboard results also expose only the combined score, which rules out any component-level comparison of Recall@5 and Macro-F1 across submissions. Taken together, the experiments in this paper support system-level comparisons: they tell us which configuration scores higher. They do not support controlled causal claims about which individual component is responsible for a given gain.

NumCoT itself is limited to surface-form extraction and pairwise comparison: it does not normalise units, identify denominators, resolve temporal scope, or check whether a source actually supports a number. Combining structured numerical representations with targeted evidence-sentence selection is one natural extension. Modelling agreement and contradiction across the trace set directly, through cross-attention or graph-based ranking, may also make better use of the candidate pool than scoring each trace independently. Finally, a controlled validation sweep over the number and quality of available traces – rather than the qualitative inspection reported in Section 7 – would let us test the aggregation strategy under deliberately corrupted or reduced trace sets.

9. Conclusion

We presented the *nadie* system for CheckThat! 2026 Task 2. The final system combines a DeBERTa-v3-large pointwise verifier, deterministic NumCoT augmentation, class-weighted BCE, and top-5 softmax-weighted voting. It obtains an official combined score of 0.422, improving over the organiser baseline of 0.408. Across our submissions, model scale produces the largest observed gain. NumCoT provides a small, low-cost improvement, while evidence-aware modelling is competitive when context is restricted. Pairwise ranking and multi-trace aggregation also improve over their simpler counterparts, but increasingly complex objectives do not automatically yield better verification.

Acknowledgments

We thank the CheckThat! 2026 organisers for maintaining the shared-task infrastructure and providing the dataset. We also thank Dr. Simon Clematide and Andrianos Michail for their guidance and feedback. Experiments were conducted on the University of Zurich cluster platform.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI ChatGPT to assist with drafting SLURM scripts for model training and inference, and to polish the language of the working notes. All AI-generated suggestions were reviewed, edited, and tested by the authors, who take full responsibility for the final content and for the execution of the experiments.

References

[1] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: Proceedings of NAACL-HLT, 2018, pp. 809–819. doi:10.18653/v1/N18-1074.

[2] V. V, A. Anand, A. Anand, V. Setty, QuanTemp: A real-world open-domain benchmark for fact-checking numerical claims, arXiv preprint arXiv:2403.17169 (2024).
[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, in: Working Notes of CLEF 2026, Jena, Germany, 2026.
[4] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, A. Mittal, FEVEROUS: Fact extraction and VERification over unstructured and structured information, in: Proceedings of the NeurIPS Datasets and Benchmarks Track, 2021.
[5] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, M. Gardner, DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs, in: Proceedings of NAACL-HLT, 2019, pp. 2368–2378. doi:10.18653/v1/N19-1246.
[6] Q. Ran, Y. Lin, P. Li, J. Zhou, Z. Liu, NumNet: Machine reading comprehension with numerical reasoning, in: Proceedings of EMNLP-IJCNLP, 2019, pp. 2474–2484. doi:10.18653/v1/D19-1251.
[7] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, in: Proceedings of ICLR, 2023.
[8] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, J. Schulman, Training verifiers to solve math word problems, arXiv preprint arXiv:2110.14168 (2021).
[9] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, K. Cobbe, Let's verify step by step, arXiv preprint arXiv:2305.20050 (2023).
[10] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, arXiv preprint arXiv:2111.09543 (2021).
[11] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150 (2020).