Overview of the CLEF-2026 CheckThat! Lab Task 2 on Fact-Checking Numerical Claims

Notebook for the CheckThat! Lab at CLEF 2026

Venktesh V[1], Vinay Setty[2], Primakov Chungkham[3], Avishek Anand[4] and Firoj Alam[5]

[1]Stockholm University
[2]University of Stavanger, Norway
[2]Independent Researcher
[4]Delft University of Technology, Netherlands
[5]Qatar Computing Research Institute, Qatar

Abstract

We present an overview of the CheckThat! Lab 2025 Task 3, part of CLEF 2026. The task focuses on verifying claims with numerical quantities and temporal expressions, leveraging Large Language Models. Numerical claims are defined as those requiring validation of explicit or implicit quantitative or temporal details. The task is conducted in a test-time scaling setup, where given a claim and evidence several reasoning paths judging the veracity of the claims are sampled from a frozen Large Language Model (LLM). The claim,evidence, reasoning path combinations are provided as samples to the participants who are expected to train an Outcome or Process verifier to rank the reasoning paths based on their utility to lead to the right verdict. The task is conducted in three languages: Arabic, Spanish, and English. A total pf 21 teams participated in English, 13 teams in Arabic, and 14 teams in Spanish. A total of 258 runs were submitted in English, 55 runs in Spanish and 86 runs for Arabic. We provide a description of the dataset, the task setup, including evaluation settings, and a brief overview of the participating systems. As is customary in the CheckThat! Lab, we release all the datasets as well as the evaluation scripts to the research community.[1] This will enable further research on identifying challenges with fact-checking numerical claims that can assist various stakeholders, such as fact-checkers, financial research analysts, and policymakers.

Keywords

Fact-Checking, Numerical Claims, Test-Time Scaling, Large Language Models, Reasoning

1. Introduction

Automated fact-checking helps combat misinformation at scale [1, 2, 3, 4], but it remains a knowledge-intensive task requiring multi-step reasoning over multiple evidence passages to arrive at a reliable verdict. Verification of claims that contain numerical information presents an additional challenge, which requires contextualizing numbers in the correct scope, around surrounding facts, as well as complex numerical reasoning to arrive at the correct verdict regarding the veracity of the claim [5]. Numerical claims are the core focus of this task as misrepresenting numerical facts can have severe consequences, for e.g., the opioid epidemic, where Purdue Pharma misrepresented numerical facts regarding the efficacy of their drugs [6]. This is partly due to the *Numeric-Truth Effect* [7], whereby the presence of numbers induces a false sense of credibility, thereby accelerating the spread of misinformation.

The CheckThat! 2026 lab was held in the framework of CLEF 2026 [8, 9].[1] Figure 1 shows the full CheckThat! identification and verification pipeline, highlighting the four tasks targeted in this ninth edition of the lab: Task 1 focused on source retrieval for scientific web claims, asking participants to identify the source publication of informal references made by users in their social media posts. Task 2 focuses on [10] addressed the challenge of verifying numerical and temporal claims, adding a fresh

[1]https://gitlab.com/checkthat_lab/clef2026-checkthat-lab/-/tree/main/task2

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

venktesh (V. V); vsetty@acm.org (V. Setty); fialam@hbku.edu.qa (F. Alam)
0000-0001-5885-2175 (V. V); 0000-0001-7172-1997 (F. Alam)

[1]https://checkthat.gitlab.io

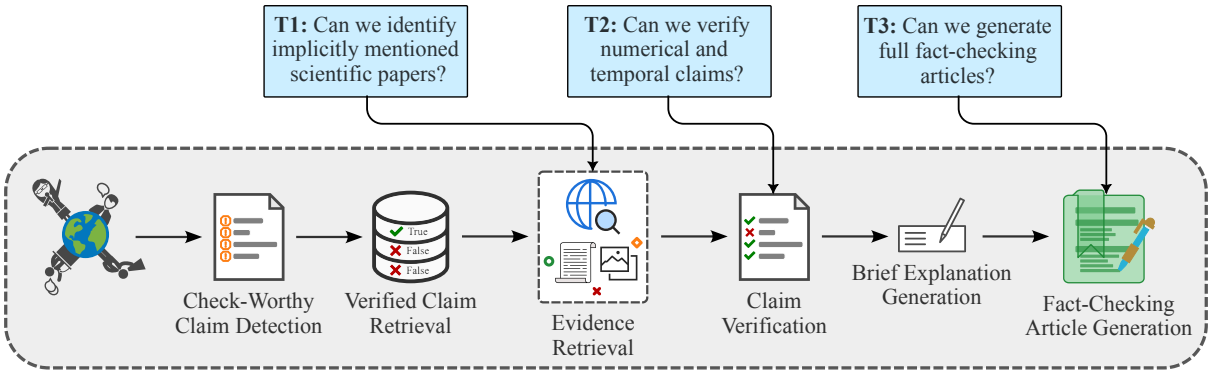

Figure 1: The CheckThat! verification pipeline, featuring the core fact-checking tasks along with the CheckThat! 2026 tasks.

reasoning component to the 2025 edition of the task (this paper). Task 3 [11] was a new addition to the CheckThat! fact-checking pipeline, which asked participants to generate full fact-checking articles with inline citations given a claim, its verdict, and the evidence for reaching the verdict..

Recent studies demonstrate that Large Language Models (LLMs) still struggle at verifying numerical claims compared to verifying other non-numerical claims [5, 12, 13]. Hence, we draw inspiration from recent advances in test-time scaling (TTS) for fact-verification [12]. In such a TTS setup, LLMs are allotted more compute during inference to generate multiple reasoning traces for claim verification. This is followed by scoring the traces using a verifier/reward model to select the best trace leading to the correct verdict. Therefore, we propose a test-time scaling setup for this task where the primary goal is to train a reward model that can help identify accurate reasoning traces by ranking them according to scores from the reward model. This is an extension of Task 3 from the 2025 edition of CheckThat!.

The remainder of the paper is organized as follows: Section 3 describes the datasets released with the task. We present the evaluation setup in Section 4. Section 5 discusses the system submissions and the official results. Section 2 presents some related work. Finally, we provide some concluding remarks in Section 6

2. Background

2.1. Fact-Checking and Challenges

Automated fact-checking is typically cast as claim verification: given a claim and a body of retrieved evidence, the system must decide whether the claim is supported, refuted, or lacks sufficient evidence [14]. A large fraction of early work adopts a modular design comprising document retrieval, evidence sentence selection, and verdict prediction, with the final step frequently instantiated as natural-language inference [15]. Such architectures perform well on straightforward claims but break down when claims are numerical or political in nature, since these demand evidence aggregation across sources together with multi-hop reasoning [16, 17]. Motivated by this shortcoming, [18] propose decomposing a complex claim into its explicit and implicit sub-questions, each answerable against the evidence, prior to rendering an overall verdict. However, these approaches still fall short on reasoning about numerical information or contextualizing and scoping the right numerical information in mixed contexts when numbers are mentioned in scope of political or other scenarios [12]. While recent advances in Large Language Models (LLMs) have shown to be quite adept at numerical reasoning, recent works [12] demonstrate that they suffer from *reasoning drift* with the inability to contextualize and map the right numerical information to the corresponding aspect of the claim. Hence, the authors propose test-time scaling (TTS) as a possible mitigation approach. An overview of TTS approaches are provided in the following section.

2.2. Test-Time Scaling for Reasoning Tasks

Test-Time Scaling (TTS) improves the reasoning capabilities of LLMs by allotting more compute for reasoning or to search through multiple reasoning traces. Methods such as Chain-of-Thought [19] and Tree-of-Thought [20] prompting encourage models to generate explicit intermediate reasoning steps, improving performance on complex tasks. When multiple reasoning traces are sampled, aggregation strategies such as majority voting through self-consistency [21] select the most reliable answer from the candidate set, substantially improving accuracy on numerical, multi-hop, and document-level benchmarks.

A key finding of [22] is that test-time compute can be more effective when allocated through search or verifier-guided selection rather than through naive scaling of sampled traces.

Hence, verifiers/ learned reward models help guide the ranking or search over the reasoning traces, where reasoning traces are scored. *Outcome reward models* (ORMs) [23] evaluate the final answer or verdict produced by a trace, while *process reward models* (PRMs) [24, 25] assess the quality of individual intermediate reasoning steps, providing finer-grained feedback than surface-level answer agreement. In verifier-guided Best-of-N (BoN) selection, multiple traces are sampled and the one with the highest reward model score is selected as the final prediction rather than relying on majority voting. [26] adopts this approach for fact-checking, where a reward model can be trained to score the consistency between a generated justification and its proposed verdict.

3. Data

Dataset overview and statistics: The data for Task 2 covers three languages: English, Spanish, and Arabic. Each sample consists of a claim, top-k corresponding evidence (with k varying from 3-10 depending on the claim), and a verdict (target label). The verdict label is one of "True", "False" or "Conflicting". The distribution of claims across different splits in datasets are shown in Table 2. The English claims from the training and development splits, along with their corresponding evidence and annotated verdicts are derived from QuanTemp [5]. The 3,260 Arabic and 2,808 Spanish claims with corresponding evidence in the training and development splits are reused from last year's iteration of the task [27].

Reasoning Traces: Each claim in the training and the validation datasets comprises 15 traces after de-duplication (20 traces for test set) generated by `gpt-4o-mini`. Each trace is generated with a maximum output length of 1024 tokens and top-p sampling with $p = 1.0$. The decoding temperature is varied across traces according to a balanced schedule ranging from 0.30 to 0.70 in steps of 0.05, as described in Section 3.1. For each claim, the most relevant reasoning traces that lead to the correct verdict (ground truth) are also annotated with an indicator that can be leveraged for training and evaluation of the reward model. To prevent data leakage from previous iterations, new test sets have been collected for each language.

3.1. Prompt Construction and Reasoning Trace Generation

First the evidence passages are retrieved using BM25 from the corpus for each claim. Then the evidence is re-ranked using a `paraphrase-MiniLM-L6-v2` sentence-transformer model [28] model for each claim. Then a prompt is constructed for each claim using the re-ranked evidence to aid the LLM in generating reasoning traces. The prompt contains the claim, the ranked evidence, and up to four semantically similar few-shot examples. The few-shot examples are selected by embedding the current claim and comparing it with claims from a held-out set of annotated examples. The nearest examples are included to demonstrate the expected reasoning format and output structure.

Generator Prompt Template

```
Here are examples of fact-checking:
[Claim]: <example claim 1>
[Subquestions + Evidence]
- Q: <sub-question 1> Evidence: <evidence doc 1>
- Q: <sub-question 2> Evidence: <evidence doc 2>
[Label]: <SUPPORTS|REFUTES|CONFLICTING>
... (further few-shot examples) ...

Following given examples, for the given claim, given
sub-questions and evidence use information from them to
fact check the claim and additionally paying attention
to numerical spans in claim and evidence and output the
label by performing entailment. Use the sub-questions to
guide your reasoning. Do not give [Label] for each
sub-question. Classify the entire claim as strictly one
of: SUPPORTS, REFUTES or CONFLICTING.

[Claim]: "<current claim>"
[Evidence]: "<current evidence>"

Generate [Justification]: and [Label]: in the end for
the whole claim. [Label]: must be strictly one of
SUPPORTS, REFUTES or CONFLICTING.
[Justification]: (in English, Spanish or Arabic depending on the split)
[Label]:
```

Each prompt instructs the generator to reason over the evidence and produce a single reasoning for the whole claim, followed by one final veracity label. The model is explicitly instructed to output the final answer using the fields `[Justification]:` and `[Label]:`. The label is constrained to one of the three supported classes: {SUPPORTS, REFUTES, CONFLICTING}. The prompt is shown above. A similar prompt is also adapted for Spanish and Arabic with claim, evidence and few shot examples provided in the corresponding language.

Reasoning traces are generated using gpt-4o-mini served through OpenAI API. For each claim, multiple traces are sampled by varying the decoding temperature. The temperature grid ranges from 0.35 to 0.65 with a step size of 0.05. This encourages diversity among the generated reasoning paths while keeping the maximum trace budget fixed.

Annotation Guidelines 3.1: Reasoning-Path Annotation

Please annotate the reasoning path for each sample provided after the prefix `Justification:` along two dimensions.

Fluency (Likert scale, 1–5). Rate grammatical fluency in the language of the justification: **1** = least fluent; **2** = sounds plausible but grammatical errors still exist; **3** = relatively better, with fewer errors; **4** = mostly correct, with only minor errors; **5** = most fluent.

Faithfulness (scale 1–5). Rate **5** if the justification is faithful to the label it predicts: if the verdict is *False*, the justification should lean towards this verdict, and likewise for *True* or *Conflicting* labels. *Conflicting* mostly occurs when some aspects of the claim are incorrect; the justification must therefore explain which aspects are correct and which are incorrect.

Annotators may refer to the separate evidences, clearly formatted in the JSON file corresponding to each claim.

An example of reasoning paths for some Spanish claims are as shown in Table 1. All reasoning paths generated were manually evaluated by experts in the respective languages. An overview of guidelines

provided to the experts is as shown in the environment - “Annotation Guidelines 3.1"

Table 1: Claims with reasoning and verdicts.

Claim	Reasoning	Verdict
ABC tuitea que cuatro de cada cinco jóvenes del colectivo LGTBI se sentirían más protegidos en un país islámico que en una España gobernada por Vox	No se ha encontrado ninguna evidencia que apoye la afirmación de que ABC haya publicado una noticia sobre “cuatro de cada cinco jóvenes del colectivo LGTBI” sintiéndose más protegidos en un país islámico que en España bajo el gobierno de Vox. La búsqueda realizada no arrojó resultados, lo que indica que tal afirmación es infundada y carece de respaldo en fuentes verificables. (REFUTES)	FALSE
Fotografía de Pablo Iglesias con un traje que le viene grande en el desfile del 12 de Octubre	La afirmación sobre la fotografía de Pablo Iglesias en el desfile del 12 de Octubre es incorrecta. No hay evidencia que respalde que su traje le venga grande; más bien, se menciona su mascarilla y su presencia en el evento. Además, los detalles sobre otras prendas que ha usado no tienen relación con la afirmación inicial sobre su vestimenta. Por lo tanto, no se puede confirmar que la descripción sea precisa. (REFUTES)	FALSE

For the training set, the generator produces up to 15 reasoning traces for each claim after deduplication, and 20 traces are generated for the test split across the three languages.

Table 2
Task 2: Claim dataset statistics

Split	English	Spanish	Arabic
Train	6,400	2,246	2,608
Dev	1,600	562	652
Test	2,558	1,164	511

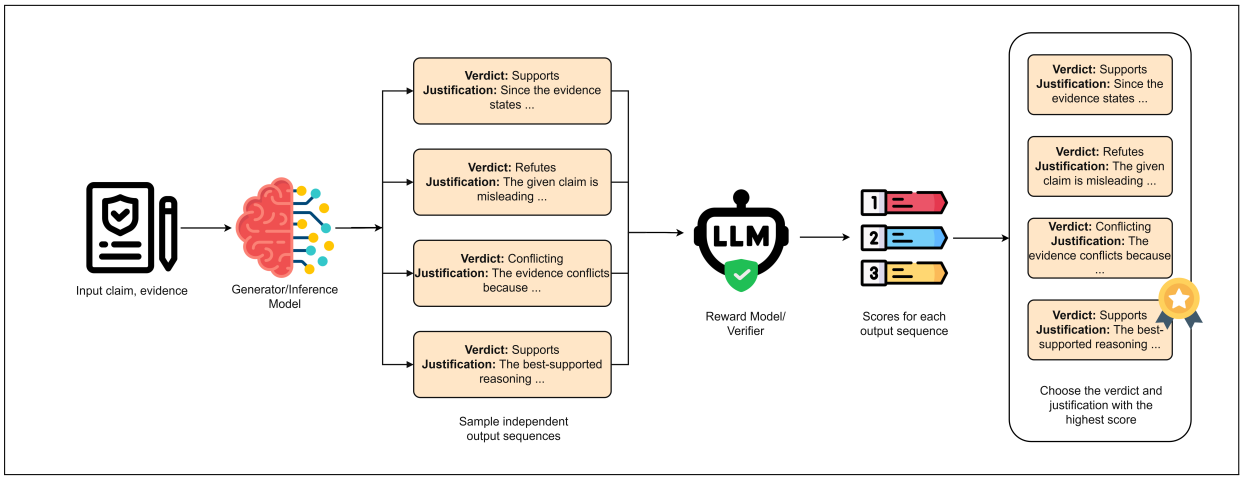

Figure 2: Overview of the generate-then-verify framework for verifier-guided Best-of-N selection.

4. Evaluation Setup

An overview of the evaluation setup is shown in Figure 2. Given a claim, evidence are retrieved and re-ranked, following by the generation of reasoning traces as explained in Section 3.1. Each claim, evidence, and reasoning trace combination is annotated with a "1" or "0" as a label, which indicates whether the reasoning trace leads to the correct verdict. Then, given the claim, evidence, and reasoning traces on the training split, participants are expected to train a verifier (ORM or PRM) that learns to score the reasoning traces. For evaluation, claims with ranked evidence and reasoning traces are provided. Participants are expected to provide a ranking of the reasoning traces by predicting a score for each trace using the verifier model, similar to the Best-of-N (BoN) setting [23, 12].

Automatic evaluation measures. Task 2 is a combination of ranking reasoning traces and deriving the right verdict based on the top-ranked reasoning trace. Hence, we use Recall@5 to measure the quality of the reasoning traces ranked by the verifier model. For claim verification, we use macro-averaged F1. Participants were judged based on high performance on both macro F1 and Recall@k. The final score used for ranking participants is obtained by averaging macro F1 and Recall@k.

5. Submissions

An overview of approaches is shown in Table 3. A total of 21 teams participated in English, 13 teams in Arabic and 14 teams in Spanish.

5.1. Overview of the Participants' Approaches

A significant number of teams focused on fine-tuning LLM-based models, like `Llama-3.1` or `Qwen`-based models as reward models for the task of scoring reasoning chains. Some teams also observed that smaller language models coupled with useful loss variants achieved comparable performance gains, resulting in more compute-efficient reward models. An overview of the different approaches adopted by different teams is provided below:

Team **Checkmates** [29] trained `Qwen2.5 (3B)` using LoRA on the given training data across the three provided languages. It was used as a ranking model by adding a custom head on top of its hidden states to produce a reasoning-quality score. They trained on multiple settings, using pointwise loss, pairwise, or listwise loss. They also employ an ensemble of rubrics-based evaluation to judge the quality of the reasoning trace, coupled with an LLM-based reward model, which led to significant gains over the individual approaches. They achieved **highest scores on two languages: AR and ES**.

Team **GippLab**[30] fine-tuned `Llama-3.1-8B-Instruct`, a transformer-based LLM, as a multilingual binary trace scorer on combined multi-lingual data using QLoRA with 4-bit loading. For each claim, the team scored candidate reasoning traces using the claim, evidence snippets, trace verdict, and cleaned justification. A trace was labeled positive when its verdict matched the gold claim label. At inference time, traces were ranked by the learned scorer. The final claim verdict was derived from the top-5 trace verdicts. Training used early stopping on macro-F1.

Team **PrompterSquad** [39] fine-tuned a `DeBERTa-v3-base` encoder as reward model which is frozen and adapted with LoRA on the query_proj, key_proj, value_proj, and dense modules for efficient training. The [CLS] pooled representation is passed through a three-layer MLP scorer, producing a single relevance score per (claim, evidence, reasoning trace) triple. They employed a hybrid loss combining MarginRankingLoss (margin=0.3, weight 0.7) with a ListMLE listwise term (weight 0.3) for training the reward model.

Team **2e1b** [34] also fine-tuned a reward model based on `Qwen3-Base`, leveraging LoRA and use Best-of-N (top-5 by score) to vote on the final verdict. The claim and reasoning trace pairs are provided as input to the reward model without evidence, as the authors observe that introducing evidence sometimes causes distraction in the reward model, leading to errors in the ranking of traces.

Table 3
Task 2: Leaderboard and overview of the approaches for fact-checking numerical claims. **Bold scores** represent best performance on that language. Lamnguages: Arabic (AR), English (EN), Spanish (ES)

Team	Language			Model												Macro-F1		
	AR	ES	EN	BM25	TF-IDF	gpt-4o-mini	Llama-3.2	Mistral	XLM-RoBERTa	ModernBERT	RoBERTa-base	Llama3.1-8B	Qwen Series	Deberta-v3-base	Logistic Regression	mDeBERTa-v3-base	AR	ES	EN
CheckMates [29]	■	■	■			■							■				**86.79**	**67.49**	59.48
GippLab [30]	■	■	■			■						■					83.40	65.08	**61.50**
Outsider	■	■	■			■		■					■				85.83	61.35	59.79
AlexU CSED [31]	■	■	■			■											84.30	59.20	59.71
Mircean2	■	■	■			■							■				86.02	66.07	60.10
DS@GT [32]	■	■	■			■	■			■							84.85	57.67	57.98
5people	■	■	■														82.68	63.82	59.04
Einon [33]	■	■	■			■					■						83.05	53.91	55.88
2e1b [34]	■	■	■			■							■				83.76	55.45	58.98
Dokyoon	■	■	■														82.93	52.35	58.00
Amora	■	■	■														79.07	56.62	60.33
SINAI-UGPLN [35]	-	■	■		■	■		■							■		-	63.42	55.15
Nadie [36]	-	-	■									■					-	-	60.11
VeriFind [37]	-	-	■				■					■					-	-	59.88
LIMBO@GT [38]	-	-	■			■	■					■					-	-	60.02
PrompterSquad [39]	-	-	■			■								■			-	-	59.67
AKSH[40]	-	-	■			■			■			■					-	-	58.17
Sshrestha	-	-	■														-	-	58.00
Overfit Udpd	-	-	■														-	-	57.49
Hajymyrat	-	-	■		■										■		-	-	57.06
Ponsubash Raj R	■	■	■														56.19	49.90	54.45

Team **Micrean2** also employed a smaller LLM (`Qwen2.5-1.5B-Instruct`) as a reward model. They augment the training process of this reward model with Camel-based preprocessing and sigmoid-weighted verification.

Team **VeriFind** [37] fine-tuned the `Llama-3.2-1B` model using LoRA rank=8, alpha=16, applied to the query, key, and value projection layers. They only trained roughly 1-2% of the model parameters, which made this memory efficient.

Team **AKSH** [40] employed a heterogeneous, tri-modal ensemble designed to verify numerical claims and rank reasoning traces (Best-of-N). To capture different dimensions of linguistic understanding, the team combined an encoder-only model (`XLM-RoBERTa`) with two decoder-only models (`Llama-3.2-1B-Instruct` and `GPT-4o`). The final predictions are generated via a weighted probability blend optimized independently for English, Spanish, and Arabic.

Team **LIMBO@GT** [38] employed an evidence-aware solution pipeline rather than a reasoning-traces-based one and fine-tuned a transformer-based reward model verifier. LIMBO@GT also employed a test-time scaling (TTS) technique to dynamically allocate computational resources during inference.

Team **Outsider** adopted an interesting alternative approach where the reward model was fine-tuned in a pure Output Reward Model (ORM) fashion, where they directly ranked claim-verdict pairs, ignoring the reasoning traces and evidence. The verifier/reward model was based on `Mistral-Small-3.1-24B-Base-2503`, fine-tuned separately for English, Arabic, and Spanish with 4-bit NF4 quantization and LoRA adapters.

Team **DS@GT** [32] framed trace ranking as a binary classification problem. Each reasoning trace was paired with its claim and assigned a binary label: 1 if the trace verdict matched the gold label, 0 otherwise. For each claim, up to six traces were sampled, roughly two matching the gold label and the rest drawn from opposing verdict types — to ensure a balanced training set. Several LLMs, namely `Llama-3.2-1B`, `ModernBERT-large`, and `Qwen2.5-Math-7B` were fine-tuned using LoRA (rank=8, alpha=16). The team observed that `Qwen2.5-Math-7B` performed best among all backbones, likely due to its mathematical reasoning pretraining.

Team **SINAI-UGPLAN** [35] explored two main families of methods: supervised thresholded verification based on TF-IDF and Logistic Regression, and confidence-aware LLM-assisted verification over uncertain examples. The final English submission, used Mistral as a non-fine-tuned verifier for selected uncertain cases and achieved an official score of 0.3952. The final Spanish submission, used supervised training and validation thresholding, achieving 0.4582. Their results show different behavior across languages: LLM-assisted verification provided a small improvement for English, whereas Spanish was better served by supervised thresholding. This suggests that numerical claim verification with reasoning traces requires language-specific calibration and that LLM-based post-processing should be applied with caution when not fine-tuned to the target language and label distribution.

Team **AlexU-CSED** [31] proposed two approaches for training a verifier to score reasoning traces generated through test-time scaling (i) a decoder-only LLM generative verifier finetuned for next-token prediction goal to score traces via Yes/No token prediction, and (ii) a discriminative verifier ensemble that fuses a reranker trained with listwise loss with a claim verifier trained with weighted cross-entropy loss via Reciprocal Rank Fusion. They evaluated approaches on three languages Arabic, English, and Spanish. On test set, our generative verifier achieves a test score of 0.5876 on Arabic leaderboard ranking 5th, and 0.4080 on English leaderboard, our reranker-verifier fusion approach scores 0.4189 on Spanish leaderboard ranking 7th. They also conducted ablation experiments on Dev sets to investigate how well model and lora rank scaling improve scores when different losses are used for training verifier.

6. Conclusion

We presented a detailed overview of Task 2 from the CheckThat! Lab at CLEF 2026. Task 2 attracted 21 teams, with English attracting the highest number of participants. Most teams followed a low-rank adaptation-based approach to fine-tune LLMs as reward models to rank the reasoning traces. A few teams also observed that smaller transformer-based encoders like ModernBERT, XLM-RoBERTa are competitive with fine-tuned LLMs, demonstrating more efficient approaches for training and inference of reward models ranking traces. Apart from focus on models, several teams also explored different loss functions to improve ranking of traces. For instance, some adopted listwise ranking losses, treating the task as a full-ranking problem, over pointwise losses. However, an analysis of the results indicates that this change does not affect final performance significantly. The performance on the leaderboard demonstrates that while TTS helps bridge the gap compared to existing verification frameworks, numerical claim verification is still far from being a solved problem demonstrating future potential for research.

Acknowledgments

The dataset generation was enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725 and Zeus cluster provided by Department of Computer and Systems Sciences

at Stockholm University. This research is also partially funded by SFI MediaFutures partners and the Research Council of Norway (grant number 309339) and Innovation Norway commercialization grant for Factiverse. The work of F. Alam was supported by NPRP grant 14C-0916-210015 from the Qatar National Research Fund, part of the Qatar Research Development and Innovation Council (QRDI).

Declaration on Generative AI

Generative AI powered tools was used only for gramatical checks and for formatting tables.

References

[1] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the association for computational linguistics 10 (2022) 178–206.

[2] V. Setty, Factcheck editor: Multilingual text editor with end-to-end fact-checking, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2744–2748.

[3] P. Nakov, D. Corney, M. Hasanain, F. Alam, T. Elsayed, A. Barrón-Cedeño, P. Papotti, S. Shaar, G. Da San Martino, Automated fact-checking for assisting human fact-checkers, in: Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI '21, 2021, pp. 4551–4558.

[4] V. V, V. Setty, Livefc: A system for live fact-checking of audio streams, in: Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 1060–1063. URL: https://doi.org/10.1145/3701551.3704128. doi:10.1145/3701551.3704128.

[5] V. Venktesh, A. Anand, A. Anand, V. Setty, Quantemp: A real-world open-domain benchmark for fact-checking numerical claims, in: 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Association for Computing Machinery (ACM), 2024, pp. 650–660.

[6] A. Van Zee, The promotion and marketing of oxycontin: commercial triumph, public health tragedy, American journal of public health 99 (2009) 221–227.

[7] N. Sagara, Consumer understanding and use of numeric information in product claims, University of Oregon, 2009.

[8] 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, The clef-2026 checkthat! lab: Advancing multilingual fact-checking, in: R. Campos, A. Jatowt, Y. Lan, M. Aliannejadi, C. Bauer, S. MacAvaney, A. Anand, Z. Ren, S. Verberne, N. Bai, M. Mansoury (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2026, pp. 325–335.

[9] 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.

[10] 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: [41], 2026.

[11] D. Sahnan, T. Chakraborty, P. Nakov, Overview of the CLEF-2026 CheckThat! lab task 3 on generating full fact-checking articles, in: [41], 2026.

[12] P. Chungkham, V. V, V. Setty, A. Anand, Think right, not more: Test-time scaling for numerical claim verification, in: Findings of the Association for Computational Linguistics: EMNLP 2025, Association for Computational Linguistics, Suzhou, China, 2025, pp. 24345–24363.

[13] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, A. Mittal, FEVEROUS: fact extraction and verification over unstructured and structured information,

in: J. Vanschoren, S. Yeung (Eds.), Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual, 2021.

[14] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819. URL: https://aclanthology.org/N18-1074/. doi:10.18653/v1/N18-1074.

[15] S. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, in: Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 632–642.

[16] Y. Jiang, S. Bordia, Z. Zhong, C. Dognin, M. Singh, M. Bansal, HoVer: A dataset for many-hop fact extraction and claim verification, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 3441–3460. URL: https://aclanthology.org/2020.findings-emnlp.309/. doi:10.18653/v1/2020.findings-emnlp.309.

[17] R. Aly, Z. Guo, M. S. 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 2015 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 632–642.

[18] J. Chen, A. Sriram, E. Choi, G. Durrett, Generating literal and implied subquestions to fact-check complex claims, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 3495–3516.

[19] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[20] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, Advances in neural information processing systems 36 (2023) 11809–11822.

[21] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, arXiv preprint arXiv:2203.11171 (2022).

[22] C. Snell, J. Lee, K. Xu, A. Kumar, Scaling llm test-time compute optimally can be more effective than scaling model parameters, arXiv preprint arXiv:2408.03314 (2024).

[23] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al., Training verifiers to solve math word problems, arXiv preprint arXiv:2110.14168 (2021).

[24] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, I. Higgins, Solving math word problems with process-and outcome-based feedback, arXiv preprint arXiv:2211.14275 (2022).

[25] 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, in: The twelfth international conference on learning representations, 2023.

[26] 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.

[27] 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. Venktesh, Overview of the CLEF-2025 CheckThat! Lab: Subjectivity, fact-checking, claim normalization, and retrieval, in: J. Carrillo-de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[28] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.
[29] H. Ghanem, M. Bassem, M. Torki, N. El-Makky, CheckMates at CheckThat! 2026: Combining Ranking Models and LLM Judges for Numerical Reasoning Trace Verification, in: [41], 2026.
[30] A. R. Khawaja, A. M. Lehavi, Gipplab at checkthat! 2026: Multilingual numerical claim verification with qlora trace scoring, in: [41], 2026.
[31] M. M. Abdelaziz, M. M. Masseoud, M. Torki, N. El-Makky, AlexU CSED Digi at CheckThat! 2026: Generative Verifier and Fused Reranker-Verifier for Fact-Checking Numerical Claims, in: [41], 2026.
[32] S. Sinha, S. Shrestha, DS@GT ARC at CheckThat! 2026: LLM-Based Trace Ranking and Grouped Reward Modeling for Multilingual Numerical Claim Verification, in: [41], 2026.
[33] J. Tang, J. Luo, K. Lin, Z. Han, Einon at CheckThat! 2026: Reward Model with Confidence-Based Filtering for Numerical Claim Verification, in: [41], 2026.
[34] H. B. Bankoglu, E. Aydin, E. Ozbaran, 2e1b at CheckThat! 2026: Comparing Pointwise, Pairwise, and Listwise Verifier Objectives Across Model Scales for Numerical Claim Verification, in: [41], 2026.
[35] M. Toapanta, M. A. Garcia, Cumbreras, L. A. Urena, Lopez, SINAI-UGPLN at CheckThat! 2026: Confidence-Aware Reasoning Trace Ranking for Numerical Claim Verification, in: [41], 2026.
[36] M. Yu, M. Emelianov, nadie at CheckThat! 2026: Numerical Reasoning-Enhanced Verifier for Fact-Checking Numerical Claims, in: [41], 2026.
[37] D. Zhang, X. Song, M. Rashid, A. S. Anik, J. Ding, L. Hong, VeriFind at CheckThat! 2026: Enhancing Numerical Claim Verification via Fine-Tuned LLMs, in: [41], 2026.
[38] L. Tao, S. Zhong, LIMBO@GT ARC at CheckThat! 2026: Ensemble Models of Evidence-aware and Verdict Candidates Detector for Numerical Claim Verification, in: [41], 2026.
[39] T. S. Pham, H. D. T. Nguyen, PrompterSquad at CheckThat! 2026: Self-Distilled Hard Negatives and a Hybrid Pairwise–Listwise Ranker for Fact-Checking Numerical Claims, in: [41], 2026.
[40] S. H. Raza, A. Karim, F. Alvi, A. Samad, AKSH at CheckThat! 2026: A Tri-Model Ensemble Approach for Fact-Checking Numerical Claims, in: [41], 2026.
[41] 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.