Team HUMANE at AVeriTeC 2025: HerO 2 for Efficient Fact Verification

Yejun Yoon[♡] **Jaeyoon Jung**[♣◇] **Seunghyun Yoon**[♠] **Kunwoo Park**[♣♡]
[♡]Department of Intelligent Semiconductors, Soongsil University
[♣]School of AI Convergence, Soongsil University
[◇]MAUM AI Inc.
[♠]Adobe Research, USA
{yejun0382, jaeyoonskr}@soongsil.ac.kr, syoon@adobe.com, kunwoo.park@ssu.ac.kr

Abstract

This paper presents HerO 2, Team HUMANE's system for the AVeriTeC shared task at the FEVER-25 workshop. HerO 2 is an enhanced version of HerO, the best-performing open-source model from the previous year's challenge. It improves evidence quality through document summarization and answer reformulation, optimizes veracity prediction via post-training quantization under computational constraints, and enhances overall system performance by integrating updated language model (LM) backbones. HerO 2 ranked second on the leaderboard while achieving the shortest runtime among the top three systems, demonstrating both high efficiency and strong potential for real-world fact verification. The code is available at https://github.com/ssu-humane/HerO2.

1 Introduction

This paper describes Hero 2, the fact verification system developed by Team HUMANE for the AVeriTeC shared task. Hero 2 is an improved version of HerO (Yoon et al., 2024), which achieved the state-of-the-art performance among the open-source models in the last year's AVeriTeC shared task. The 2025 edition emphasizes efficient, reproducible, and open-source approaches to automated fact-checking. Two key changes distinguish this year's task setting. First, computational and time constraints prohibit the use of large language models with more than ten billion of parameters (e.g., Llama3 70B Instruct). Second, the evidence evaluation metric has shifted from Hungarian METEOR (a token-based metric) to Ev2R recall (a model-based metric), which requires the generation of more flexible and semantically coherent evidence.

In alignment with these goals, Hero 2 enhances retrieval performance through document-based retrieval and summarization, and reconstructs answer texts based on the question. We further improve the verification process using AWQ (Lin et al., 2024), enabling higher accuracy while maintaining efficiency under the hardware constraints specified by the task. As a result, Hero 2 achieved second place on the leaderboard while exhibiting the shortest runtime among the top-performing models, demonstrating its efficiency and suggesting its potential for real-world fact verification.

2 Task Description

The AVeriTeC shared task aims to build a fact-checking system that verifies real-world claims using web evidence. The claim verification process consists of three main steps. First, the system performs evidence retrieval by collecting relevant web documents. Next, during question generation, the system may generate questions for each piece of evidence to better assess the claim, though this step is optional. Finally, in the veracity prediction phase, the system uses the collected information to assess the truthfulness of the claim. The possible verdicts are: supported, refuted, not enough evidence, or conflicting evidence/cherry-picking.

The 2025 shared task specifically targets two main goals. First, it aims to promote the development of high-performing systems that, using only open LLMs, can retrieve relevant evidence and generate accurate verdicts to maximize evaluation scores. Second, it emphasizes the importance of building reproducible and efficient fact-verification systems. Accordingly, all systems must be executable within the provided virtual machine environment and capable of verifying a single claim in under one minute. While the previous shared task (Schlichtkrull et al., 2024) used the Hungarian METEOR score to assess the quality of questions (Q score) and question-answer pairs (Q+A score), this year's evaluation adopts the Ev2R recall (Akhtar et al., 2024), an LLM-based evaluation method. Ev2R utilizes an LLM to decompose the

Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER), pages 224–228
July 31, 2025 ©2025 Association for Computational Linguistics

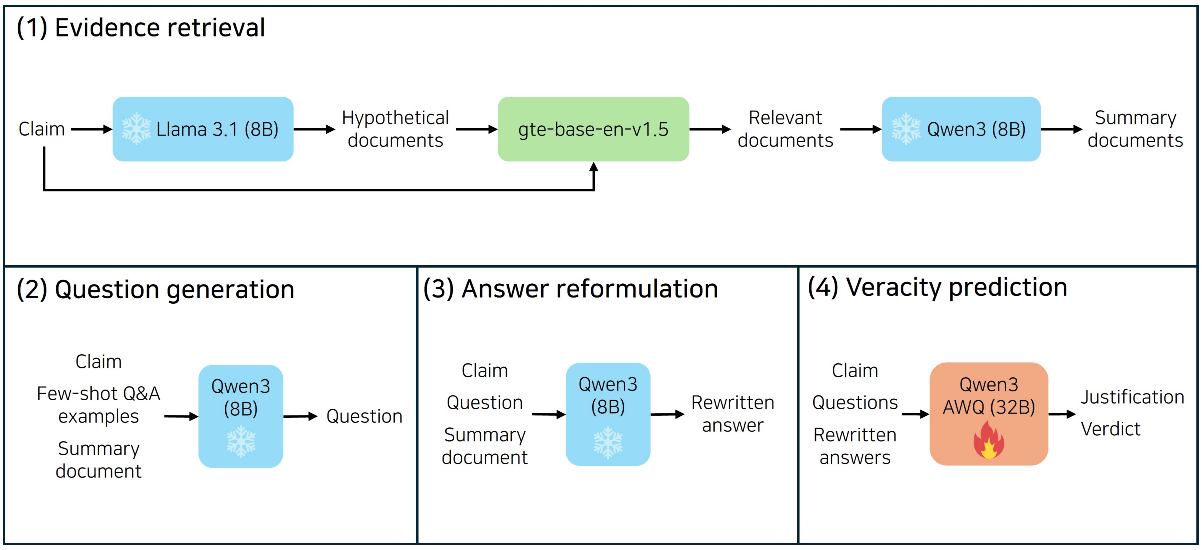

Figure 1: Pipeline of our system

System	Query Expansion	Evidence Retrieval	Evidence Summarization	Question Generation	Answer Reformulation	Veracity Prediction
Baseline	HyDE-FC (Llama3.1 8B)	Hybrid (BM25/SFR-embedding-2)	NA	Llama3 8B	NA	Llama3.1 8B
HerO 2		Dense (gte-base-en-v1.5)	Qwen3 8B	Qwen3 8B	Qwen3 8B	Qwen3 32B AWQ

Table 1: Model configurations

ground-truth evidence into atomic facts. The Q+A score is then calculated by measuring the degree to which these facts cover the predicted evidence. The new AVeriTeC score is computed as the veracity prediction accuracy when the Q+A score of the predicted evidence for a claim exceeds a predefined threshold.

3 Our System: HerO 2

We present HerO 2, an improved fact verification pipeline of HerO (Yoon et al., 2024). The key enhancements are summarized below:

- **Document summarization**: Web documents are summarized into paragraph-level evidence blocks.
- **Answer reformulation**: A language model is prompted to convert the retrieved evidence blocks into answer-form texts.
- **Post-training quantization**: A fine-tuned LLM is quantized for veracity prediction.
- **Updated LM backbones**: Backbone LMs for each components are updated to maximize the performance.

Figure 1 illustrates its overall pipeline, and Table 1 details the model configuration in comparison to the baseline method.

3.1 Knowledge Store Construction

The 2025 AVeriTeC Shared Task imposes a one-minute time limit for processing each claim on a designated virtual machine. To meet this constraint, we apply two preprocessing steps to the web documents provided as the knowledge store: (1) indexing dense embeddings for all documents using gte-base-en-v1.5 (Li et al., 2023) following the design choice of the winning model in last year's shared task (Rothermel et al., 2024); and (2) summarizing each document into paragraph-level evidence candidates using Qwen3 8B (Yang et al., 2025).

3.2 Evidence Retrieval

The goal of evidence retrieval is to retrieve evidence necessary for verifying claims from the knowledge store. We use HyDE-FC (Yoon et al., 2024) for query expansion, which generates hypothetical fact-checking articles for a given claim by prompting an LLM. We retrieve the top 10 relevant documents through the indexed dense embedding. We adopt an additional step to summarize each of

> **Your task is to read the following document carefully and summarize it into a single, coherent paragraph. Focus on capturing the main ideas and essential details without adding new information or personal opinions.**
>
> **Document:**: An image of a purported 1998 letter from actor Sean Connery (famous for his portrayal of agent James Bond) to Apple CEO Steve Jobs, caustically rebuffing an offer to become a pitchman for Apple Computers, hit the Internet in June 2011. ...
>
> **Output**: In June 2011, an image of a purported 1998 letter from actor Sean Connery to Apple CEO Steve Jobs, rejecting an offer to become an Apple pitchman, circulated online. However, the letter was part of a satirical article published on Scoopertino, a website known for fabricating news about Apple under the motto "All the News That's Fit to Fabricate." ...

Figure 2: An example of the instruction prompt used for document summarization, along with its output. The bold text is the instruction, and the blue text indicates the model output.

> **The following text provides an evidence obtained through web searches related to a specific question, used for verifying the accuracy of a claim. Your task is to answer the question based on this evidence. Ensure your answer is Supported by relevant context from the evidence.**
>
> **Claim**: In a letter to Steve Jobs, Sean Connery refused to appear in an apple commercial.
>
> **Question**: Was the letter from Sean Connery to Steve Jobs genuine?
>
> **Evidence**: In June 2011, an image of a purported 1998 letter from actor Sean Connery to Apple CEO Steve Jobs, rejecting an offer to become an Apple pitchman, circulated online. However, the letter was part of a satirical article published on Scoopertino, a website known for fabricating news about Apple under the motto "All the News That's Fit to Fabricate." ...
>
> **Answer**: No, the letter from Sean Connery to Steve Jobs was not genuine. It was part of a satirical article published on Scoopertino, a website known for fabricating news about Apple. ...

Figure 3: An example of the instruction prompt used for answer reformulation, along with its output. The bold text is the instruction, and the blue text indicates the model output.

the retrieved documents into a single paragraph. The used prompt for summarization is shown in Figure 2. Our best model uses Llama3.1 8B for HyDE-FC and Qwen3 8B for document summarization.

3.3 Question Generation and Answer Reformulation

In this step, we generate questions based on the summarized evidence and retain only the information necessary to answer them through sequential LLM generations. We first generate questions using the same prompt as the baseline method. Then, we transform the retrieved summary text into an answer-form response conditioned on the claim and the generated question. The prompt used for this reformulation step is shown in Figure 3. Our best-performing pipeline employs Qwen3 8B for both question generation and evidence reformulation.

3.4 Veracity Prediction

We use a fine-tuned instruction-following language model to predict the veracity of given claims. Our best-performing model is a fine-tuned Qwen3 32B model, quantized to 4-bit using AWQ (Lin et al., 2024) to satisfy the VRAM constraints of the shared task, enabling inference on an A10G GPU (23GB). Following the baseline approach, we use a prompt that incorporates the annotator's rationale into the veracity prediction process. The top 10 question–answer pairs generated in the earlier stages are provided as in-context examples, along with the claim to be verified.

4 Evaluation Experiments

This section presents the experimental results that guided the selection of each module in the submitted system.

4.1 Experimental Setups

In the comparison experiments, we used the development set to evaluate model performance. We used the training set for training our models. The Ev2R evaluation is carried out in a local environment using the Llama 3.3 70B (Grattafiori et al., 2024) model with a threshold of 0.5.

For retrieval, we used gte-base-en-v1.5, which supports a context length of 8192. In other cases, we employed mxbai-embed-large-v1 (Lee et al., 2024). All language models used in the experiments were instruction-tuned versions. For training, we used the Adam optimizer with a learning rate of 2e-5, batch size of 128, and trained the model for 2 epochs.

The Llama3.1 8B (Grattafiori et al., 2024) was configured with the following hyperparameters for HyDE-FC: maximum number of tokens as 512, temperature as 0.7, and top-p as 1.0. The Qwen3 8B used the hyperparameters recommended by the Qwen team[1]: temperature as 0.7, top-p as 0.8, top-k as 20, and min-p as 0. Veracity predictions used the hyperparameters: temperature as 0.9, top-p as 0.7, and top-k as 1. If the language model failed to produce a verdict label, we repeated the generation using top-2 sampling.

We ran experiments using three machines. The first has two H100 GPUs (80GB per GPU) and 480GB RAM. The second has eight H100 GPUs with 2TB RAM; the third has four NVIDIA A6000 GPUs (48GB per GPU) and 256GB RAM. The experiments were conducted in a computing environment with the following configuration: Python 3.12.9, PyTorch 2.6.0, Transformers 4.51.3, vLLM 0.8.5, and Sentence-Transformers 4.1.0.

4.2 Experimental Results

Evidence Retrieval We compare three retrieval strategies: (1) retrieving individual sentences, (2) retrieving consecutive sentence chunks with one-sentence overlap, and (3) retrieving entire documents. Table 2 presents the evidence retrieval results on the development set, varying the number of retrieved evidence candidates. Among the three, the top-10 document-level retrieval strategy achieves the best performance, outperforming all other configurations with different retrieval targets and candidate counts.

Retrieval Target	Q + A (Ev2R recall)		
	Top-3	Top-5	Top-10
Sentence	0.289	0.315	0.374
Chunk (2 sentences)	0.311	0.369	0.404
Chunk (3 sentences)	0.347	0.382	0.413
Chunk (4 sentences)	0.364	0.41	0.115
Document	**0.37**	**0.487**	**0.522**

Table 2: Evidence retrieval performance

Answer Reformulation Table 3 presents the results of document summarization and answer reformulation, varying the number of retrieved evidence documents used for veracity prediction. The best performance is observed when answer reformulation is applied to the top-10 question-answer pairs. When controlling for the number of retrieved documents, applying answer reformulation consistently outperforms the baseline without reformulation.

Method	Q + A (Ev2R recall)		
	Top-3	Top-5	Top-10
Document-based retrieval	0.37	0.487	0.522
+ document summarization	0.483	0.51	0.487
+ answer reformulation	**0.501**	**0.514**	**0.556**

Table 3: Effects of answer reformulation

Veracity Prediction Controlling for other modules by fixing them to their best-performing configurations, we compare veracity prediction methods using the optimal settings for evidence retrieval and question generation. Table 4 presents the fine-tuning results. While the models achieve comparable F1 scores, they exhibit substantial differences in accuracy. Since the AVeriTeC score is based on accuracy, we adopt it as the primary metric for selecting the best-performing model. Notably, applying AWQ post-training quantization to Qwen3 32B yields a +0.018 improvement in accuracy over its computationally equivalent smaller variant, Qwen3 8B. Among the 8B models, Qwen3 outperforms Llama 3.1, the language model used in the baseline.

Method	F1	ACC
Qwen3 32B AWQ	0.382	**0.692**
Qwen3 8B	**0.385**	0.674
Llama3.1 8B	0.384	0.588

Table 4: Veracity prediction performance

4.3 Test Set Results

System	AVeriTeC score	Average runtime per claim (s)
CTU AIC	**0.332±0.002**	53.67
HerO 2	0.271±0.004	**29.19**
yellow_flash	0.253±0.005	31.71
Baseline	0.202±0.007	33.88

Table 5: Test set results

Table 5 presents the performance of HerO 2 on the test set in comparison with the baseline and other competitive models. CTU AIC achieved the highest AVeriTeC score of 0.332, followed by HerO 2 with 0.271 and yellow_flash with 0.253. HerO 2 also achieved the lowest average runtime per claim at 29.19 seconds, demonstrating superior efficiency compared to CTU AIC (53.67 seconds), yellow_flash (31.71 seconds), and the baseline (33.88 seconds). These results indicate that HerO 2 is the most efficient system among the top-performing models. It is worth noting that while some systems

[1] https://huggingface.co/Qwen/Qwen3-8B

achieved shorter runtimes, their performance was significantly lower.

5 Conclusion

In this paper, we presented HerO 2, an efficient fact-checking system developed for the AVeriTeC shared task, hosted by the eighth FEVER workshop. Four key components contribute to the performance improvement of HerO 2: document summarization, answer reformulation, post-training quantization, and updated language model backbones. Our system achieved second place in the shared task while recording the shortest runtime among the top three systems, highlighting its efficiency and potential for real-world fact verification.

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization (IITP-2025-RS-2022-00156360) and Graduate School of Metaverse Convergence support (IITP-2025-RS-2024-00430997) programs, supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). Yejun Yoon and Jaeyoon Jung contributed to this work equally as co-first authors. Kunwoo Park is the corresponding author.

References

Mubashara Akhtar, Michael Schlichtkrull, and Andreas Vlachos. 2024. Ev2r: Evaluating evidence retrieval in automated fact-checking. *arXiv preprint arXiv:2411.05375*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. Open source strikes bread - new fluffy embeddings model.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Mark Rothermel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. InFact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112, Miami, Florida, USA. Association for Computational Linguistics.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos. 2024. The automated verification of textual claims (AVeriTeC) shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yejun Yoon, Jaeyoon Jung, Seunghyun Yoon, and Kunwoo Park. 2024. HerO at AVeriTeC: The herd of open large language models for verifying real-world claims. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 130–136, Miami, Florida, USA. Association for Computational Linguistics.