

AEScorer: An Agentic Evidence-Grounded Framework for Graded Factuality Verification

Hui Huang^{1,2}, Muyun Yang¹, Yuki Arase²

¹Harbin Institute of Technology, ²Institute of Science Tokyo
huanghui@stu.hit.edu.cn, arase@c.titech.ac.jp

Abstract

Despite the significant advancements of Large Language Models (LLMs), their factuality remains a critical challenge, creating a growing need for more nuanced factuality verification. Existing factuality verification methods do not capture graded judgments, even though factuality is better understood as a spectrum rather than a binary of right and wrong. To bridge this gap, we focus on graded factuality verification and propose **AEScorer**, an agentic evidence-grounded framework with two stages: agentic evidence acquisition and graded scoring. AEScorer first gathers and refines external evidence through agentic search, and then predicts a scalar factuality score to distinguish nuanced differences in factual correctness. We further construct **GradedVeriBench**, a benchmark for graded factuality verification spanning both general and multi-hop question answering. Experimental results on GradedVeriBench show that AEScorer substantially outperforms existing methods across both settings, demonstrating the value of coupling targeted evidence acquisition with graded scoring¹.

1 Introduction

Large Language Models (LLMs) such as GPT-4 have achieved remarkable progress in natural language generation and are now widely used in many applications (Zhao et al., 2026; Liu et al., 2025). However, they can still generate plausible-sounding but factually incorrect content (Wang et al., 2023). This phenomenon, commonly referred to as factual hallucination, remains a major barrier to deploying LLMs in scenarios where factual errors have practical consequences.

Existing factuality verification methods mostly operate in coarse categories such as correct or incorrect (Lin et al., 2022a). Recent work has discussed *fine-grained* factuality verification focusing on the

¹Code and data are openly available at <https://github.com/HuihuiChyan/FactVeri-SFT>.

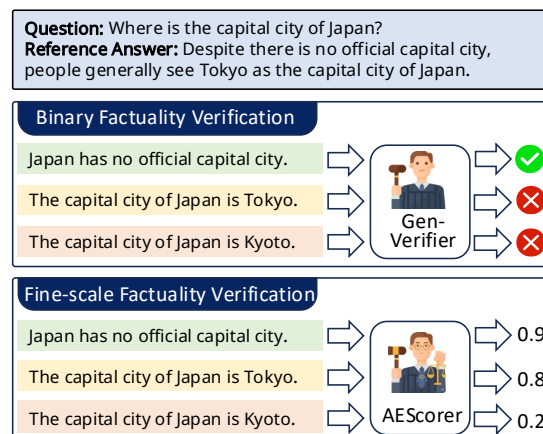


Figure 1: Illustrative comparison between coarse factuality verification and graded factuality verification. A single atomic fact may contain different degrees of factual error that binary verification cannot distinguish.

granularity of texts to verify and employed a segmented atomic claim as an evaluation unit, rather than a response (Min et al., 2023; Mitra et al., 2025). However, a short-form response or a single atomic fact can be factually correct to different degrees, as illustrated in Figure 1.

We refer to this problem as *graded factuality verification*: the task of distinguishing different degrees of factual incorrectness within a single atomic fact. This task is important in preference alignment applications, where reward models and reranking pipelines benefit from dense and informative factuality signals. It is also theoretically well motivated, as epistemological perspectives characterize truth as graded belief rather than a strict binary switch (Ramsey, 1926), while fuzzy logic formalizes truth values on a continuous interval (Zadeh, 1965). These perspectives support treating factuality verification as a graded assessment problem rather than only a binary decision problem.

To address this task, we propose **AEScorer**, an agentic evidence-grounded framework for graded

factuality verification. AEScorer first performs *agentic evidence acquisition*, where the model interactively retrieves and refines external evidence for the target answer. It then conducts *graded scoring*, predicting a scalar score that reflects nuanced factual differences rather than only coarse labels. AEScorer conditions its scoring on evidence retrieved through agentic search, enabling evidence-supported graded factuality scores at inference time. By training the scoring function using pairwise preferences, AEScorer avoids the need for absolute score annotations (Kasa et al., 2025).

We further introduce **GradedVeriBench**, a benchmark designed for graded factuality verification targeted at short-form question answering. GradedVeriBench is constructed through a ranking-based annotation pipeline that combines LLM-as-a-Judge (Calderon et al., 2025) with human verification to ensure reliable nuanced factuality judgments. Experimental results show that AEScorer substantially outperforms existing verification methods on GradedVeriBench and transfers to various verification scenarios.

Our primary contributions are threefold:

1. We formalize *graded factuality verification*, which aims to distinguish different degrees of factual incorrectness within a single fact.
2. We propose **AEScorer**, an agentic evidence-grounded framework that couples evidence acquisition with graded scoring.
3. We introduce **GradedVeriBench**, the first benchmark for graded factuality verification. Experiment results on GradedVeriBench demonstrate that AEScorer is superior to existing methods in terms of graded verification.

2 Related Work

2.1 Factuality Verification

In the context of LLMs, factuality verification is the process of evaluating whether a piece of model-generated text, such as a claim or response, is factually accurate (Wang et al., 2023). It also serves as a central mechanism for hallucination detection (Huang et al., 2025).

Reliable benchmarks are fundamental for measuring factuality verification progress. Early works such as TruthfulQA (Lin et al., 2022b) and HaluEval (Li et al., 2023a) treat factuality mainly as a binary problem, providing foundational datasets for evaluating whether LLMs can recognize hal-

lucinations and assess overall factual correctness. However, coarse labels are often insufficient for applications that require nuanced factuality feedback.

To address the complexity of longer generations, subsequent research has shifted toward fine-grained assessment. FactScore (Min et al., 2023) popularized the *Decompose-Then-Verify* paradigm, which decomposes long-form responses into atomic facts for individual verification. Follow-up works, including FELM (Zhao et al., 2023) and WiCE (Kamoi et al., 2023), further refined fine-grained verification over decomposed claims. Although this line of work has refined the *unit of evaluation*, the resulting labels for each unit often remain binary or coarse. A graded benchmark that supports nuanced scoring within a single atomic fact remains absent.

2.2 LLM-as-a-Verifier

LLM-based factuality verifiers can be broadly grouped into two technical paradigms: generative verifiers and discriminative verifiers.

Generative verifiers leverage the generation ability of LLMs to retrieve external knowledge, produce reasoning traces, and output a final factuality assessment, represented by work such as FactScore and VeriScore (Song et al., 2024). Recent methods have extended this paradigm in different directions: FGLR (?) enhances NLI-based reasoning by generating auxiliary premise facts, while FineSumFact (Oh et al., 2024) uses fine-grained LLM feedback to supervise factuality in summarization. However, prompting generative models to output numerical scores does not directly optimize them for scoring, leading to poorly calibrated graded scores.

In contrast, discriminative verifiers assess factuality more directly by mapping the input to a scalar factuality score or class label. For instance, Paudel et al. (2025) use a classification head to validate responses against context and common knowledge. Similarly, Li et al. (2024) propose a lightweight classifier for factual consistency evaluation. This formulation enables fast and efficient optimization for fine-grained scoring (Wang et al., 2024). However, without an explicit mechanism for acquiring external evidence, discriminative verifiers are often constrained by knowledge insufficiency.

3 Proposed Method: AEScorer

Graded factuality verification is challenging because it requires both sufficient evidence grounding and sensitive score calibration. To address

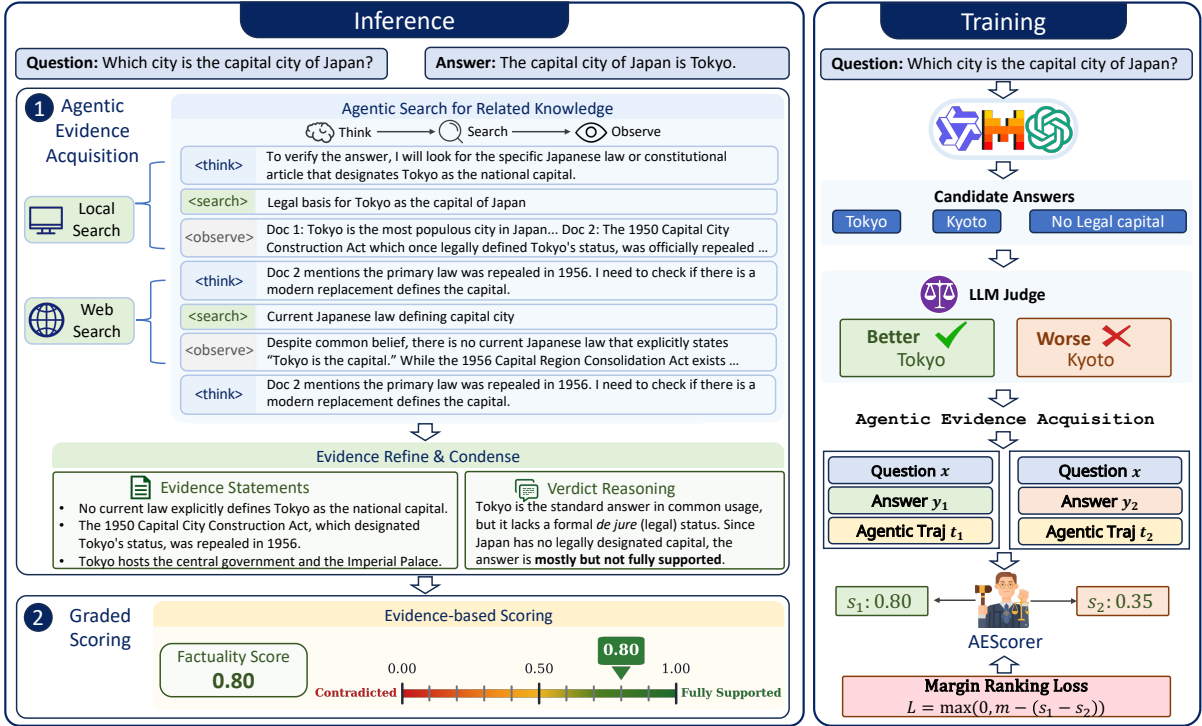


Figure 2: The overall pipeline of graded factuality verification with AEScorer. The left side illustrates unified inference with agentic evidence acquisition and graded scoring, while the right side shows pairwise data construction and the training process for its evidence-based scoring capability.

these two requirements, we propose AEScorer, which treats evidence acquisition and scoring as two tightly connected stages. As illustrated in Figure 2, it first performs *Agentic Evidence Acquisition* to collect and refine verification-oriented evidence, and then applies *Graded Scoring* to produce a scalar factuality score grounded on the evidence.

3.1 Agentic Evidence Acquisition

Agentic Search Without access to external information, a verifier may suffer from severe knowledge gaps, especially when the target answer concerns temporal, entity-centric, or multi-hop facts. Therefore, AEScorer employs an agentic search mechanism (Jin et al., 2025) that allows the model to plan evidence-seeking actions and autonomously retrieve evidence whenever its internal knowledge is insufficient for verification.

The retrieval process follows an iterative loop with the following steps:

1. **Think:** AEScorer plans the next verification action and formulates a targeted search query based on the missing evidence.
2. **Search:** AEScorer retrieves information from external tools, including WebSearch for broad and up-to-date online evidence and

LocalSearch for a trusted document repository.

3. **Observe:** The retrieved results are incorporated into the running context to support the next round of reasoning. The loop terminates once AEScorer determines that the available evidence is sufficient for scoring.

This process produces an evidence-rich trajectory containing both retrieved knowledge and intermediate reasoning, establishing the factual grounding required for graded verification. In addition, the iterative loop allows AEScorer to adapt its search to the specific missing facts in the target answer.

Evidence Refinement Raw search trajectories are often verbose, noisy, and redundant. Since LLM-based systems operate under a finite attention budget and suffer from context decay (Mei et al., 2025), AEScorer refines the acquired evidence before scoring. Concretely, it compresses the search trajectory into two compact components:

1. A list of useful evidence statements extracted from retrieved content;
2. The reasoning traces necessary for the final verification decision.

By supplying the scorer with concise, high-signal input, we prevent the scoring stage from being dominated by irrelevant trajectory details.

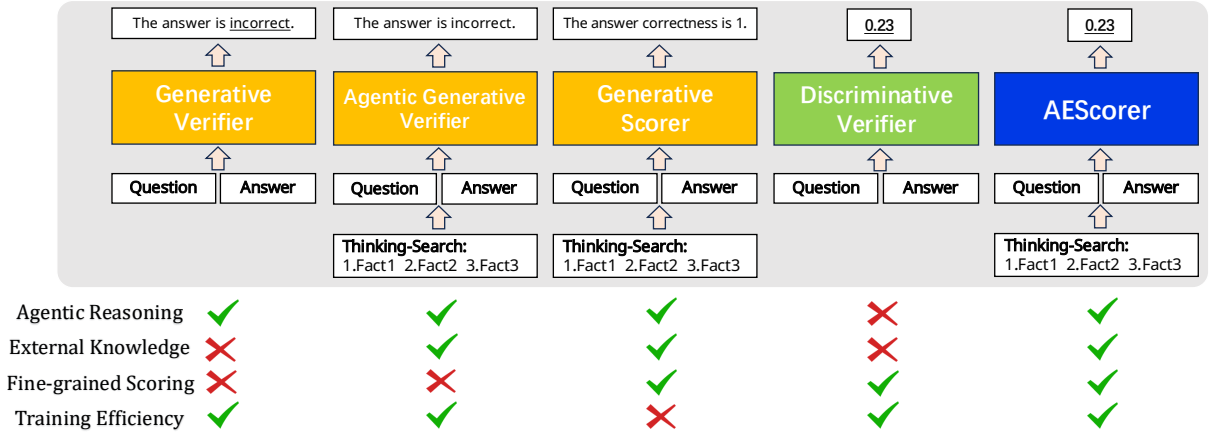


Figure 3: Comparison of different verification paradigms. We compare them in terms of agentic reasoning ability, accessibility to external knowledge, graded scoring ability, and training efficiency.

3.2 Graded Scoring

Given the refined evidence, AEScorer predicts a scalar factuality score for the target answer. Unlike coarse verification, the objective here is not merely to classify an answer as correct or incorrect, but to distinguish nuanced differences in factual correctness for a single atomic fact or short response. To support such nuanced judgments, we implement the scoring component as a regression-style head over the model representation, allowing the framework to output continuous values rather than vocabulary-constrained discrete labels (Llama, 2023). This design makes the final decision directly optimized for scoring rather than dependent on the calibration of generated text.

While our goal is to predict a standalone scalar factuality score for each answer at inference time, training on absolute scores is challenging because human annotation of absolute factuality scores is difficult and can result in inconsistency. Therefore, we train AEScorer with pairwise preferences, which provide more reliable supervision for learning the scoring function as described in Section 3.3.

3.3 Scoring Optimization

We construct training triplets of (*question*, *preferred answer*, *dispreferred answer*), as shown on the right side of Figure 2. We optimize the scoring component with a margin ranking loss:

$$L = \max(0, m - (f(x, y_+, t_+) - f(x, y_-, t_-))) \quad (1)$$

Here, $f(x, y, t)$ denotes the score assigned by AEScorer to answer y for question x given refined

evidence trajectory t , y_+ is the preferred (more factual) answer, y_- is the dispreferred (less factual) answer, t_+ and t_- denote the refined evidence trajectories for the two answers obtained from the first stage, and m is the ranking margin.

Pairwise Data Construction Directly annotating absolute scalar factuality scores for training is costly and often inconsistent. Therefore, we collect pairwise comparisons between candidate answers, which provide more reliable supervision while still enabling AEScorer to learn a standalone scoring function.

1. **Answer Generation:** Generate multiple diverse candidate answers for each question.
2. **Initial Assessment:** Use an advanced LLM-as-a-judge to evaluate each answer against a reference, assigning an initial quality label.
3. **Pair Sampling:** Randomly sample pairs consisting of a preferred (more factual) and a dispreferred (less factual) answer based on the labels.
4. **Preference Verification:** Re-check the pairwise factuality order with the judge model.
5. **Trajectory Construction:** Run the agentic evidence acquisition stage for each answer and store its refined evidence trajectory.

By training on this dataset with the margin ranking loss, AEScorer learns to assign higher scores to answers that are more factual with respect to reference information. Consequently, during inference, the trained verifier can predict a scalar score for a single, standalone answer, although supervision is provided through pairwise preferences².

²Notice this score is a discriminative relative-ranking signal rather than a probabilistically calibrated absolute metric.

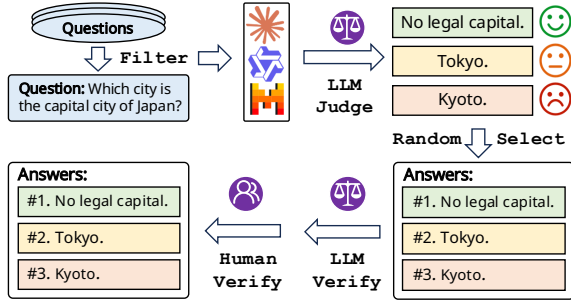


Figure 4: Construction process of GradedVeriBench.

Parameter Efficiency To ensure practical applicability and minimize storage overhead, we employ Low-Rank Adaptation (LoRA) to train the scoring capability (Hu et al., 2021). This parameter-efficient fine-tuning strategy ensures that the storage footprint of the final scorer is only marginally larger than that of the base generative model. Furthermore, it enhances the method’s adaptability across multi-domain scenarios (Tian et al., 2025).

3.4 Comparison to Existing Paradigms

As summarized in Figure 3, existing verifiers have distinct limitations as follows.

1. **Naive Generative Verifier** relies solely on LLMs’ internal knowledge and fails when claims exceed the model’s knowledge boundary.
2. **Agentic-Generative Verifier** such as Jin et al. (2025), has access to external knowledge but can not generate fine-grained scores.
3. **Agentic-Generative Scorer** such as Li et al. (2023b), is constrained by scalar score generation ability and the lack of absolute scoring data for effective optimization.
4. **Naive Discriminative Verifier** such as Liu et al. (2024), is easy to be optimized for scoring with a pairwise objective but lacks external knowledge.

AEScorer addresses these shortcomings by embodying the idea of *agentic search as feature engineering*. It combines interactive evidence gathering with a ranking-trained scoring stage, enabling both efficient training and evidence-based scoring.

4 GradedVeriBench

Existing factuality benchmarks typically focus on either coarse verification or long-form claim decomposition. To address the missing setting of *graded factuality verification*, we introduce GradedVeriBench, a benchmark designed for evaluating

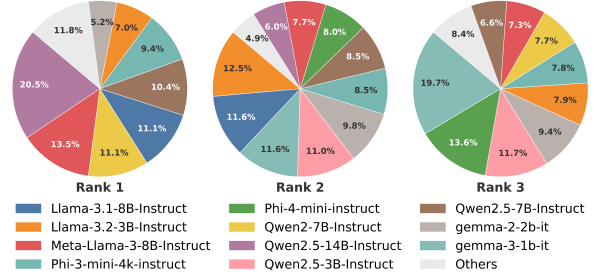


Figure 5: Model distribution of different answer ranks in GradedVeriBench. Rank 1 has the best factuality.

NQ		TriviaQA		PopQA	
Number	Length	Number	Length	Number	Length
203	15.7	372	13.7	365	14.3
HotpotQA		Musique		2wiki	
Number	Length	Number	Length	Number	Length
287	16.1	154	21.0	298	14.9

Table 1: Data statistics of GradedVeriBench.

relative factual differences among short-form answers, as shown in Figure 4.

Since determining the relative order of responses is significantly more reliable than assigning absolute pointwise scores, GradedVeriBench adopts a ranking-based annotation framework to improve data quality. This design converts nuanced factuality evaluation into a controlled ranking problem. The construction process is as follows:

1. **Instruction Filtering:** Select only verifiable questions that possess a single, unambiguous correct answer to ensure objective evaluation.
2. **Answer Pool Generation:** Leverage representative LLMs to generate a diverse pool of candidate answers for each question.
3. **Initial Verification:** Leverage a proprietary judge to assign an initial category to each answer: Correct, Incorrect or Intermediate.
4. **Random Selection:** Randomly sample three answers for each question with one from each of correct, intermediate and incorrect ones.
5. **LLM&Human-based Verification:** Leverage both an LLM and human evaluators to verify the relative order of factuality from best to worst among the three sampled answers. Three graduate-student annotators independently judge whether each LLM-proposed ranking is valid.

Let N denote the number of samples sent for human verification, we define agreement rate as

Method	Features				General QA Verification						Average	
					NQ		TriviaQA		PopQA			
	AR	EK	GS	TE	P@1	K-Tau	P@1	K-Tau	P@1	K-Tau	P@1	K-Tau
MiniCheck w/ Flan-T5 (Tang et al.)					67.98	59.28	68.82	62.19	56.99	47.40	64.60	56.29
MiniCheck w/ MiniCheck-7B (Tang et al.)					71.92	66.50	74.46	69.89	67.40	57.44	71.26	64.61
HHEM-2.1-Open (Li et al.)					66.01	56.98	69.35	60.04	57.53	45.57	64.30	54.20
FactScore w/ GPT-4 (Min et al.)					77.64	70.74	70.00	70.11	70.14	61.32	72.59	67.39
VeriScore w/ GPT-4o (Song et al.)					78.46	79.68	84.41	81.81	78.57	76.67	80.48	79.39
Results on Qwen-2.5-7B-Instruct.												
Generative Verifier	✓	✗	✗	✓	71.92	62.56	69.81	61.37	71.15	58.42	70.96	60.78
Agentic-Generative Verifier	✓	✓	✗	✓	77.83	71.43	80.91	74.73	81.10	76.26	79.95	74.14
Agentic-Generative Scorer	✓	✓	✓	✗	79.31	75.70	78.98	72.69	81.37	73.33	79.89	73.91
Discriminative Verifier	✗	✗	✓	✓	67.49	56.32	75.81	63.80	73.42	61.10	72.24	60.41
AEScorer	✓	✓	✓	✓	84.24	82.92	90.86	88.53	90.14	85.21	88.41	85.55
w/o compression	✓	✓	✓	✓	81.28	77.67	83.87	79.39	83.56	79.91	82.90	78.99
Results on Qwen-3-4B-Instruct.												
Generative Verifier	✓	✗	✗	✓	71.57	63.45	70.19	58.96	68.07	57.24	69.94	59.88
Agentic-Generative Verifier	✓	✓	✗	✓	77.34	71.10	81.99	76.16	82.47	76.99	80.60	74.75
Agentic-Generative Scorer	✓	✓	✓	✗	86.21	79.31	78.98	72.69	81.37	73.33	82.19	75.11
Discriminative Verifier	✗	✗	✓	✓	67.49	55.01	72.04	54.84	66.85	52.51	68.79	54.12
AEScorer	✓	✓	✓	✓	85.22	81.28	88.98	84.23	88.77	83.20	87.66	82.90
w/o compression	✓	✓	✓	✓	79.31	78.98	87.90	83.69	87.67	82.10	84.96	81.59

Table 2: Experimental results of different verifiers on general QA verification (AR: Agentic Reasoning, EK: External Knowledge, GS: Graded Scoring, TE: Training Efficiency)

Method	Features				Multi-Hop QA Verification						Average	
					HotpotQA		Musique		2wiki			
	AR	EK	GS	TE	P@1	K-Tau	P@1	K-Tau	P@1	K-Tau	P@1	K-Tau
MiniCheck w/ Flan-T5 (Tang et al.)					56.99	47.40	46.10	41.56	51.34	48.10	51.48	45.69
MiniCheck w/ MiniCheck-7B (Tang et al.)					62.72	50.99	53.25	46.75	58.05	53.69	58.01	50.48
HHEM-2.1-Open (Li et al.)					52.61	33.80	50.65	46.32	58.72	47.43	53.99	42.52
FactScore w/ GPT-4 (Min et al.)					73.21	54.51	72.34	69.78	68.95	62.46	71.50	62.25
VeriScore w/ GPT-4o (Song et al.)					78.66	72.63	73.23	70.54	69.69	67.82	73.86	70.33
Results on Qwen-2.5-7B-Instruct.												
Generative Verifier	✓	✗	✗	✓	63.99	41.96	57.14	48.92	48.15	24.13	56.43	38.34
Agentic-Generative Verifier	✓	✓	✗	✓	72.13	60.51	57.79	53.25	67.11	49.22	65.68	54.33
Agentic-Generative Scorer	✓	✓	✓	✗	75.26	59.12	73.38	61.90	66.11	42.73	71.58	54.58
Discriminative Verifier	✗	✗	✓	✓	63.76	45.18	64.29	51.95	59.40	32.89	62.48	43.34
AEScorer	✓	✓	✓	✓	83.28	77.24	79.87	77.92	76.51	68.68	79.89	74.61
w/o compression	✓	✓	✓	✓	72.82	64.46	65.58	58.01	65.10	51.68	67.83	58.05
Results on Qwen-3-4B-Instruct.												
Generative Verifier	✓	✗	✗	✓	55.39	41.76	59.85	50.36	52.07	31.72	55.77	41.28
Agentic-Generative Verifier	✓	✓	✗	✓	78.05	66.78	66.23	58.44	71.14	50.56	71.81	58.59
Agentic-Generative Scorer	✓	✓	✓	✗	80.28	69.95	68.18	65.37	72.20	55.71	73.55	63.68
Discriminative Verifier	✗	✗	✓	✓	58.54	43.55	61.04	40.69	59.60	34.90	59.73	39.71
AEScorer	✓	✓	✓	✓	87.80	80.49	81.82	73.59	81.21	69.35	83.61	74.48
w/o compression	✓	✓	✓	✓	83.62	73.75	80.52	72.73	72.82	66.44	78.99	70.97

Table 3: Experimental results of different verifiers on multi-hop QA verification.

$$\text{Agreement rate} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\text{all three annotators give the same judgment}). \quad (2)$$

This quantity measures unanimity when validating the LLM-proposed rankings in the pre-filtering pool. On GradedVeriBench, the agreement rate among the annotators was 64.8%, showing that nuanced factuality annotation is non-trivial.

GradedVeriBench is constructed from six question-answering (QA) datasets, categorized as

follows: 1) General QA: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023). 2) Multi-hop QA: HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), and Musique (Trivedi et al., 2022). This enables comprehensive evaluation of graded factuality verification in both general and reasoning-intensive scenarios. We also present data statistics in Figure 5 and Table 1, demonstrating diverse model sources and balanced domain coverage³.

³Two data samples are presented in Appendix G.

Evaluation Metrics Performance on this benchmark is evaluated using **Kendall’s τ** correlation coefficient to assess the global alignment with human rankings, and **Precision@1** to verify the model’s ability to identify the most factual response.

5 Experiments

We conduct a comprehensive evaluation of AEScorer on GradedVeriBench, together with ablation experiments on various verification scenarios.

5.1 Settings

Because there is no existing study that combines agentic search and graded factuality verification, we compare AEScorer with prevalent factuality verification methods as baselines: FactScore (Min et al., 2023), MiniCheck (Tang et al., 2024), HHEM (Li et al., 2024), and VeriScore (Song et al., 2024).

We also compare AEScorer with the methods discussed in Section 3.4 as an ablation study. For training the scoring capability of AEScorer and the Discriminative Verifier, we merge the training sets of NQ and HotpotQA to form a corpus following Jin et al. (2025), using the pipeline illustrated in Figure 2. Other verifiers directly use the generative ability of LLMs for verification. The base models are Qwen2.5-7B-Instruct and Qwen3-4B-Instruct (Yang et al., 2025), both with strong capabilities in reasoning and tool use. For retrieval, we employ Google Search for WebSearch and Wikipedia dumps dated up to 2018 for LocalSearch (see Appendix A for more details).

5.2 Main Results

As shown in Tables 2 and 3, the performance of both generative and discriminative verifiers drops significantly without the aid of agentic search. This is because the static, parametric knowledge within LLMs is limited and cannot be updated over time. This limitation is particularly pronounced given the relatively small size of the backbone models.

Moreover, based on the same agentic search process, AEScorer significantly outperforms its generative counterparts⁴. Although we adjust the prompts of generative baselines to score individual responses or directly output rankings, their generation-oriented objectives still hinder precise graded verification. In contrast, our optimized scorer trained with a ranking objective offers better

⁴A case study of the different verifiers is in Appendix E.

Selector	Model	Token-level F1 Score		
		NQ	Test Musique	Bamboogle
	Meta-Llama-3-8B-Inst	23.82	11.14	15.02
FactScore	GPT-4	25.81	10.44	19.05
AEScorer	Qwen-2.5-7B-Inst	24.38	14.60	23.40
AEScorer	Qwen-3-4B-Inst	26.14	14.28	21.39
	Llama-3.1-8B-Inst	19.22	9.81	13.72
FactScore	GPT-4	19.72	11.36	13.42
AEScorer	Qwen-2.5-7B-Inst	22.23	11.78	17.58
AEScorer	Qwen-3-4B-Inst	22.15	11.61	19.01

Table 4: Best-of-N ($N = 16$) selection results on question answering tasks with different verifiers.

separation between answers with nuanced factual correctness.

Furthermore, removing the evidence refinement step leads to a substantial drop in AEScorer’s performance, particularly in multi-hop scenarios. This validates the role of trajectory refinement in preserving high-signal evidence for effective scoring.

Other competing methods also underperform because they either lack dedicated optimization for graded scoring or are primarily designed for long-form verification. Despite its comparatively small backbone size, AEScorer achieves the best performance thanks to its evidence-grounded scoring design, showing that task-specific supervision can compensate for model scale in this setting.

5.3 Application to LLM Generation

Graded verification is highly valuable for improving downstream generation factuality because it can rank candidate answers by factual quality rather than merely filter obviously incorrect ones. To demonstrate this, we use AEScorer for best-of- N selection in generation. Specifically, we employ Meta-Llama-3-8B-Instruct and Llama-3.1-8B-Instruct to generate multiple candidate answers for three distinct QA datasets. The verifiers then select the most factual response from the candidates.

The F1 score on question answering test sets is shown in Table 4. The results confirm that AEScorer improves factuality for LLM generation, demonstrating the usefulness of graded verification for reranking. In contrast, FactScore struggles to separate nuanced factual differences despite using a much larger model, likely because it was designed around long-form claim decomposition rather than graded short-answer factuality.

5.4 Adaptability to Binary Verification

While our focus is graded factuality verification, we also investigate AEScorer’s effectiveness in the

Method	Model	TriviaQA		HotpotQA	
		ACC	F1	ACC	F1
FactScore	GPT-4	59.95	52.85	61.85	55.21
G-Verifier	Qwen2.5-7B-Inst	66.14	64.82	43.53	42.49
AG-Verifier	Qwen2.5-7B-Inst	75.98	74.89	50.20	49.76
AEScorer	Qwen2.5-7B-Inst	88.58	88.53	70.98	70.88
G-Verifier	Qwen3-4B-Inst	66.38	65.45	45.29	44.85
AG-Verifier	Qwen3-4B-Inst	74.41	73.66	60.39	60.51
AEScorer	Qwen3-4B-Inst	88.19	88.17	72.94	72.46

Table 5: Results on binary factuality verification

Method	Model	FactScore Dataset	
		Precision@1	Kendall-tau
FactScore	GPT-4	60.62	45.00
G-Verifier	Qwen2.5-7B-Inst	33.12	5.83
AG-Verifier	Qwen2.5-7B-Inst	46.25	25.83
AEScorer	Qwen2.5-7B-Inst	59.75	38.36
G-Verifier	Qwen3-4B-Inst	41.79	9.45
AG-Verifier	Qwen3-4B-Inst	60.51	41.40
AEScorer	Qwen3-4B-Inst	65.41	47.59

Table 6: Results on long-form factuality verification.

binary verification setting required in conventional scenarios. We reconstructed GradedVeriBench with binary annotations (correct or incorrect)⁵. We then evaluated whether different verifiers could assign higher scores to correct examples than to incorrect ones. As shown in Table 5, AEScorer achieves superior performance, demonstrating that evidence-grounded scoring also transfers well to binary factuality verification.

5.5 Application to Long-Form Verification

In this section, we apply AEScorer to long-form texts, another scenario where factuality verification is desired. For this, we use the dataset from FactScore (Min et al., 2023) which contains three long-form, model-generated responses for each question, annotated with factuality scores. Following the FactScore protocol, we first decomposed each response into atomic claims and then verified each claim individually. As shown in Table 6, AEScorer also achieves the best long-form verification performance, outperforming FactScore despite using substantially smaller LLMs than GPT-4. This suggests that evidence-grounded graded scoring remains effective even when verification is applied after claim decomposition.

⁵Given that questions in GradedVeriBench include ground-truth answers, classifying answers as correct or incorrect was straightforward for human annotators.

Method	Model	Knowledge	Average
D-Verifier	Qwen2.5-7B-Inst	-	51.87
AEScorer	Qwen2.5-7B-Inst	WebSearch	79.46
AEScorer	Qwen2.5-7B-Inst	LocalSearch	74.32
AEScorer	Qwen2.5-7B-Inst	both	80.08
D-Verifier	Qwen3-4B-Inst	-	46.92
AEScorer	Qwen3-4B-Inst	WebSearch	77.05
AEScorer	Qwen3-4B-Inst	LocalSearch	70.77
AEScorer	Qwen3-4B-Inst	both	78.69

Table 7: Kendall-tau correlation coefficient of different knowledge sources on GradedVeriBench.

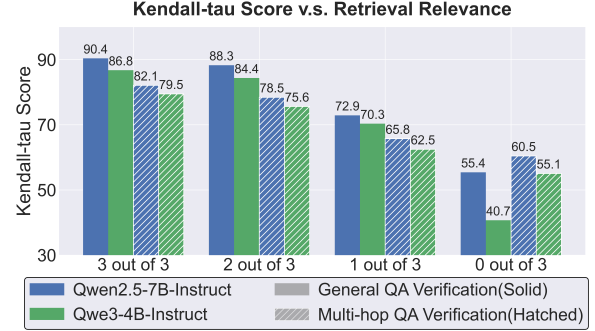


Figure 6: Impact of retrieval relevance on verification performance on GradedVeriBench. The x-axis represents the relevance level, defined as the number of candidate answers (0 to 3) supported by the retrieved context. The results demonstrate a clear trend that higher retrieval relevance leads to more accurate verification.

5.6 The Influence of Knowledge Source

As detailed in Section 3.1, our agentic search incorporates two distinct knowledge sources: WebSearch and LocalSearch. This design enables the agent to adaptively retrieve information from the most appropriate source. To validate this, we conducted ablation studies on these sources.

As shown in Table 7, WebSearch significantly outperforms LocalSearch. This result aligns with intuition, as search engines provide broad, diverse, and real-time web content while leveraging advanced algorithms for precise retrieval. In contrast, LocalSearch relies on simple similarity matching over static and limited documents. Nonetheless, combining both sources yields the best performance, demonstrating the effectiveness of diverse knowledge sources for agentic search.

To further isolate the impact of external knowledge, we use GPT-4 to assess the relevance of the retrieved results for each sample. We then categorized samples based on their retrieval relevance levels⁶, as shown in Figure 6. The results reveal a

⁶The relevance level is defined by the number of candidate

strong correlation between retrieval relevance and final verification accuracy. This confirms the critical role of external knowledge: when retrieved information is irrelevant, verification accuracy declines sharply due to the lack of reliable evidence.

6 Conclusion

Graded factuality scoring is important because it provides denser factuality feedback for answers with various degrees of correctness. We propose AEScorer for graded factuality verification. AEScorer combines agentic evidence acquisition with graded scoring, enabling grounded verification for short answers. We also develop Graded-VeriBench, a benchmark dedicated to this setting. Experimental results suggest that factuality evaluation can move beyond coarse correctness labels toward feedback signals that better reflect the graded nature of real-world factual errors.

Limitation

Our work currently has a few limitations that outline directions for future research:

- **Agentic Capabilities:** While we leveraged the inherent agentic search capabilities of Qwen-2.5-7B and Qwen-3-8B, we have not yet specifically optimized this behavior for factuality verification. Future work will focus on further enhancing and fine-tuning these search abilities.
- **Model Architecture:** Although LoRA reduces parameter overhead, the current implementation of AEScorer still leaves room for tighter integration between evidence acquisition and scoring. Future work may further simplify the pipeline to reduce computational cost and latency.
- **Application in Post-training:** Although graded verification is promising for reward modeling and reranking, its downstream value in full post-training pipelines still requires broader validation across more models and tasks.

Ethics Statement

This work uses publicly available QA benchmarks and focuses on evaluating factuality verification systems rather than generating new factual claims. To build the benchmark, we relied on a combination of LLM-based judging and human verification. The human annotation effort was conducted by graduate-student annotators following explicit answers to which the retrieved results are regarded as relevant.

guidelines, and such annotation should be fairly compensated under local research standards.

Our framework also inherits potential biases from its evidence sources and underlying models. Web search results, Wikipedia-based local corpora, and judge models may reflect historical, cultural, or distributional biases, which can affect factuality assessments. Therefore, AEScorer should be viewed as a decision-support tool rather than an unquestionable oracle, especially in high-stakes scenarios. We do not recommend using it as the sole basis for unsupervised decision making in domains such as medicine, law, or public policy.

Acknowledgments

This work was supported by National Natural Science Foundation of China (62276077) and JST K Program Grant Number JPMJKP24C3, Japan. We also thank Lvyuan Han for his assistance in implementing the agentic retrieval system.

References

- Nitay Calderon, Roi Reichart, and Rotem Dror. 2025. The alternative annotator test for llm-as-a-judge: How to statistically justify replacing human annotators with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16051–16081.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM transactions on information systems*, 43(2):1–55.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.
- Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.
- Siva Rajesh Kasa, Karan Gupta, Sumegh Roychowdhury, Ashutosh Kumar, Yaswanth Biruduraju, Santhosh Kumar Kasa, Pattisapu Nikhil Priyatam, Arindam Bhattacharya, Shailendra Agarwal, and Vijay Huddar. 2025. Generative or discriminative? revisiting text classification in the era of transformers. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9615–9637.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6449–6464.
- Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024. [HHEM-2.1-Open](#).
- Zizhong Li, Haopeng Zhang, and Jiawei Zhang. 2023b. A revisit of fake news dataset with augmented fact-checking by chatgpt. *arXiv preprint arXiv:2312.11870*.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022b. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.
- Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.
- Wei Liu, Wanyao Shi, Zijian Zhang, and Hui Huang. 2024. Hit-mi&t lab at semeval-2024 task 6: Deberta-based entailment model is a reliable hallucination detector. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1788–1797.
- Hugo Touvron Llama. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2407.21783*.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 9802–9822.
- Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. 2025. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 12076–12100.
- Kushan Mitra, Dan Zhang, Sajjadur Rahman, and Estevam Hruschka. 2025. Factlens: Benchmarking fine-grained fact verification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18085–18096.
- Jihwan Oh, Jeonghwan Choi, Nicole Hee-Yeon Kim, Taewon Yun, and Hwanjun Song. 2024. Learning to verify summary facts with fine-grained llm feedback. *arXiv preprint arXiv:2412.10689*.
- Bibek Paudel, Alexander Lyzhov, Preetam Joshi, and Puneet Anand. 2025. Hallucinator: Hallucination detection through context and common knowledge verification. *arXiv preprint arXiv:2504.07069*.
- Frank P. Ramsey. 1926. Truth and probability. In *The Foundations of Mathematics and Other Logical Essays*. Routledge and Kegan Paul, London.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

- Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. Veriscore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474.
- Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847.
- Yimin Tian, Bolin Zhang, Zhiying Tu, and Dianhui Chu. 2025. Adapters selector: Cross-domains and multi-tasks lora modules integration usage method. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 593–605.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.
- Haiyang Wang, Yuchen Pan, Xin Song, Xuechen Zhao, Minghao Hu, and Bin Zhou. 2024. F2rl: factuality and faithfulness reinforcement learning framework for claim-guided evidence-supported counterspeech generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4457–4470.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.
- Lotfi A. Zadeh. 1965. Fuzzy sets. *Information and Control*, 8(3):338–353.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, et al. 2026. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627.
- Yiran Zhao, Jinghan Zhang, I Chern, Siyang Gao, Pengfei Liu, Junxian He, et al. 2023. Felm: Benchmarking factuality evaluation of large language models. *Advances in Neural Information Processing Systems*, 36:44502–44523.

A Implementation Details

As mentioned in Section 3.3, we train the scoring capability of AEScorer with LoRA to reduce storage consumption. The training was conducted on 8 NVIDIA A6000-48GB GPUs with Transformers (Wolf et al., 2019) and DeepSpeed (Rasley et al., 2020). Detailed hyperparameters are presented in Table 8. To adapt the base model into a factuality scorer, the prediction head is newly initialized as a linear projection layer⁷.

Configuration	Qwen2.5-7B-Inst	Qwen3-4B-Inst
ranking margin	0.1	0.1
max length	1024	1024
learning rate	2e-4	1e-4
scheduler	cosine decay	cosine decay
optimizer	AdamW	AdamW
training epochs	3	3
lora rank	8	8
lora alpha	32	32
lora dropout	0.05	0.05
data types	bf16	bf16
ZeRO optimizer	stage 2	stage 2

Table 8: Settings of scoring optimization.

For retrieval on local documents, we use the 2018 Wikipedia dump (Karpukhin et al., 2020) as the knowledge source and E5 (Wang et al., 2022) as the retriever. We follow Lin et al. (2022b) and set the number of retrieved passages to 3 across all retrieval-based methods. For web retrieval, we use the Google Search API provided by Serper⁸, and we keep the top 10 snippets from the search results.

B Scaling the Scoring Component

This section examines whether the scoring component requires a large model once verification-oriented evidence has already been acquired. We vary the model size used for the scoring component in AEScorer while fixing the evidence acquisition backbone at 7B, and compare this trend with verifiers of different model sizes.

As shown in Figure 7, the generative verifier’s performance consistently improves with scorer size, as the same model must conduct reasoning, tool use, and verdict generation. The naive discriminative verifier’s performance also improves with model size, as it relies solely on internal knowledge for scoring, which scales with model capacity. In contrast, AEScorer remains relatively stable across

⁷Please refer to `AutoModelForSequenceClassification` in Huggingface library for more details.

⁸<https://serper.dev>

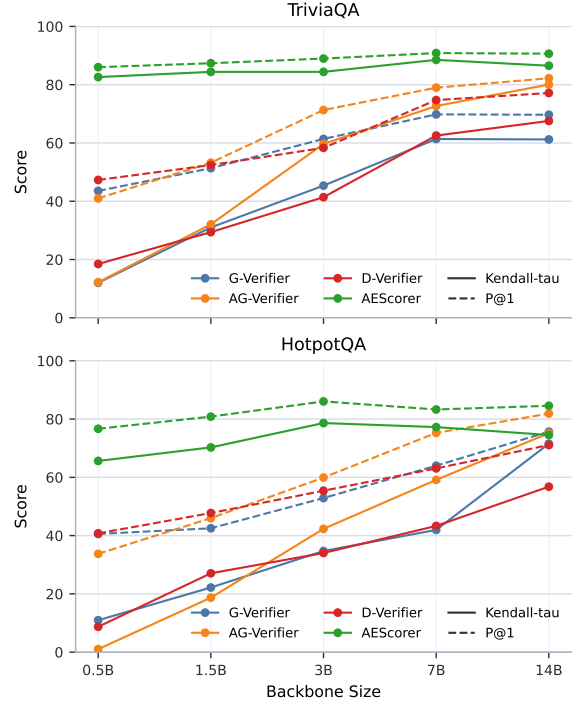


Figure 7: Variation of Kendall-tau and Precision@1 scores with different verifier sizes on TriviaQA (top) and HotpotQA (bottom).

different scorer sizes. Notably, AEScorer with a smaller scoring component still outperforms the largest AG-Verifier (14B), suggesting that explicit evidence grounding can reduce the scale requirement of the final scorer.

Moreover, the benefit of retrieval for generative verification also becomes larger as model size increases. Smaller models often lack sufficient planning and tool-use capabilities to retrieve useful evidence, whereas larger models can better exploit retrieved contexts during verdict generation.

C Scaling the Acquisition Component

Complementary to Appendix B, this section analyzes the effect of scaling the evidence acquisition backbone. We keep the scorer fixed and vary only the model used for agentic evidence acquisition, so that the experiment isolates how acquisition quality affects graded factuality scoring.

As shown in Figures 8, 10 and 9, verification accuracy drops sharply when the acquisition backbone is too small, such as 0.5B. This degradation occurs because small models often lack the planning, tool-use, and reasoning capabilities required for effective agentic search. As a result, they may retrieve incomplete evidence, fail to connect multi-hop facts, or introduce noisy trajectories that

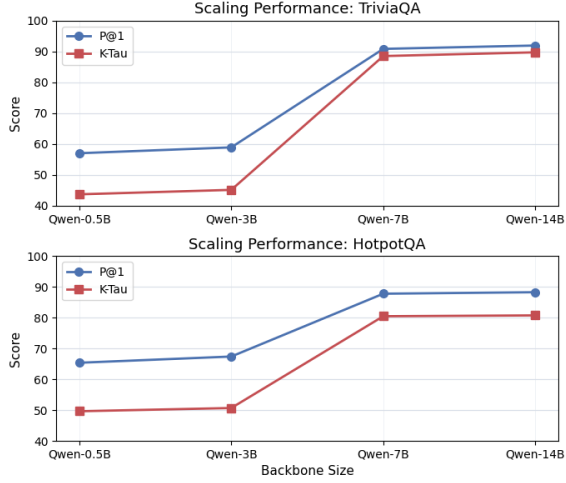


Figure 8: The influence of scaling the evidence acquisition backbone on verification accuracy.

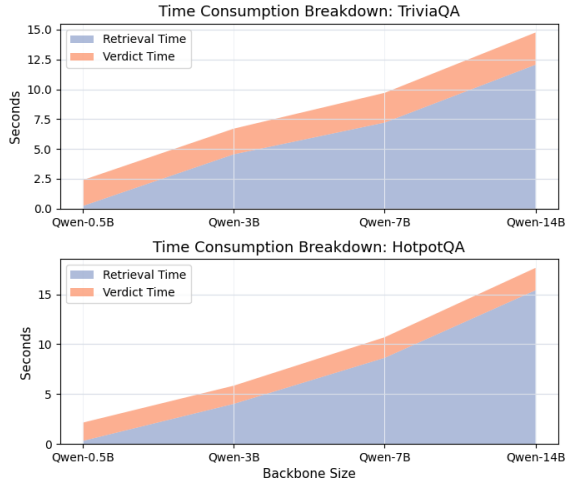


Figure 9: The influence of scaling the evidence acquisition backbone on time consumption.

weaken the downstream scoring step.

Moreover, although performance generally improves with a stronger acquisition model, larger backbones also increase latency and token consumption. The gains from 7B to 14B are marginal, suggesting a ceiling effect: once the model can reliably collect the necessary external evidence, further scaling is limited by evidence availability rather than model capacity. Therefore, the 7B acquisition backbone offers a favorable balance between verification accuracy and operational efficiency.

These results support a practical design principle for AEScorer: model capacity is more critical for evidence acquisition than for the final scoring step. An alternative implementation could therefore use a stronger model for acquisition and a lighter model for scoring, akin to the design of Tang et al. (2024). We leave this direction to future work.

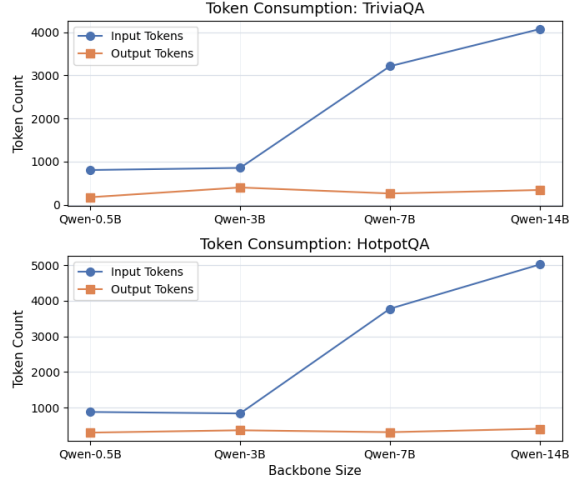


Figure 10: The influence of scaling the evidence acquisition backbone on token consumption.

D Efficiency Analysis

To better understand the computational trade-offs of agentic evidence acquisition, we compare AEScorer against the Discriminative Verifier (D-Verifier), Generative Verifier (G-Verifier), and Agentic-Generative Verifier (AG-Verifier). We report token usage, runtime, retrieval iterations, and verification accuracy for a single response.

As shown in Table 9, D-Verifier is the fastest, as it involves neither generation nor retrieval processes. The G-Verifier also maintains a relatively high speed because it requires only a single generation pass per input without retrieval. However, both models suffer from poor verification accuracy due to their lack of external knowledge.

Compared with AG-Verifier, AEScorer achieves higher verification accuracy with lower token overhead and shorter runtime. Since the two methods share the same acquisition process, this gap mainly comes from the final verdict stage: AEScorer replaces free-form verdict generation with a directly optimized scoring step. This makes the final decision both faster and more precise for graded factuality verification.

E Case Study

In this section, we conduct a qualitative case study to compare the G-Verifier with AEScorer. We selected two samples from the Natural Questions (Kwiatkowski et al., 2019) and Musique datasets (Trivedi et al., 2022), representing general QA and multi-hop QA verification respectively.

As illustrated in Figure 11 and 12, the Generative Verifier relies solely on parametric memory,

Dataset	Verifier	Tokens-Input	Tokens-Output	Time-Total	Time-Retrieval	Time-Verdict	Retrieval-Iter	P@1	K-Tau
TriviaQA	D-Verifier	472	0	0.02	0	0.02	0	75.81	63.80
	G-Verifier	472	306	5.66	0	5.66	0	69.81	61.37
	AG-Verifier	8676	527	15.82	7.21	8.61	3.02	78.98	72.69
	AEScorer	4884	365	9.70	7.21	2.49	3.02	90.86	88.53
HotpotQA	D-Verifier	478	0	0.02	0	0.02	0	63.76	45.18
	G-Verifier	478	335	6.71	0	6.71	0	63.99	41.96
	AG-Verifier	9773	586	16.98	8.63	8.35	3.56	75.26	59.12
	AEScorer	5611	420	10.69	8.63	2.06	3.56	87.80	80.49

Table 9: Efficiency comparison of different verifiers. AEScorer retains the same retrieval cost as AG-Verifier but substantially reduces the cost of the final verdict stage.

a limitation that frequently leads to factual hallucinations. For instance, in Figure 11, the model fails to adhere to strict terminological constraints, incorrectly favoring the Greek god "Hades" over the correct Roman equivalent "Dis Pater." Moreover, in Figure 12, the model suffers from knowledge conflation, hallucinating "Ponte Vecchio" (a bridge in Florence) as the answer for a query about Venice. These failures demonstrate the unreliability of relying solely on internal model knowledge for fact-checking.

Conversely, as shown in Figure 13 and 14, AEScorer leverages reasoning and tool use to acquire necessary knowledge from both local documents and web search engines. This mechanism compensates for internal knowledge deficits and provides grounded context for graded scoring. Moreover, AEScorer is particularly advantageous for multi-hop QA because it can perform targeted decomposed searches. Rather than solving the complex query in a single pass, the agent first identifies the composer, then the birthplace, and finally the bridge, thereby grounding the final verification in retrieved facts.

F Failure Analysis

Although agentic search generally improves verification, AEScorer remains sensitive to the relevance and correctness of retrieved evidence. To demonstrate this, We manually inspect the errors made by the Qwen2.5-7B-Instruct-based AEScorer on HotpotQA and identify a representative failure caused by partially related but misleading retrieval results.

Question: What year did the father of Willem van Oldenbarnevelt die?

Answer to verify: The father of Willem van Oldenbarnevelt died in 1619.

Search trace summary: The model first searched for "father of Willem van Oldenbarnevelt." The retrieved local document discussed Reinier van Oldenbarnevelt and an assassination attempt related to his father. It then searched for "Reinier van Oldenbarnevelt death year," retrieving results

that mentioned 1623 and 1 October 1636.

Compressed verdict: The model treated Reinier as Willem’s father, concluded that the evidence did not support 1619, and predicted *Incorrect*.

The retrieved information concerns Reinier van Oldenbarnevelt, who was Willem’s brother rather than his father. Because the evidence was lexically related but referred to the wrong family relation, the refinement stage retained incorrect useful facts, which then propagated to the final score. This case illustrates that retrieval does not guarantee reliable verification. Instead, partially relevant but misleading evidence can be especially harmful when the scorer lacks sufficient prior knowledge.

G Data Sample of GradedVeriBench

In this section, we present two representative samples from GradedVeriBench. As illustrated in Figures 15 and 16, the dataset exhibits a strict factuality hierarchy: Rank 1 represents the highest level of factual accuracy, Rank 2 exhibits moderate factuality, and Rank 3 incorporates severe factual errors. This design supports robust evaluation of graded factuality verification.

H Prompt templates

In this section, we present the detailed prompt templates for both the generative baselines and our proposed AEScorer. The prompts for Step 1 and Step 2 of AEScorer are presented in Prompt H.1 and H.2, while the prompts for the Generative Verifier and Generative Scorer are presented in Prompt H.3, H.4, and H.5.

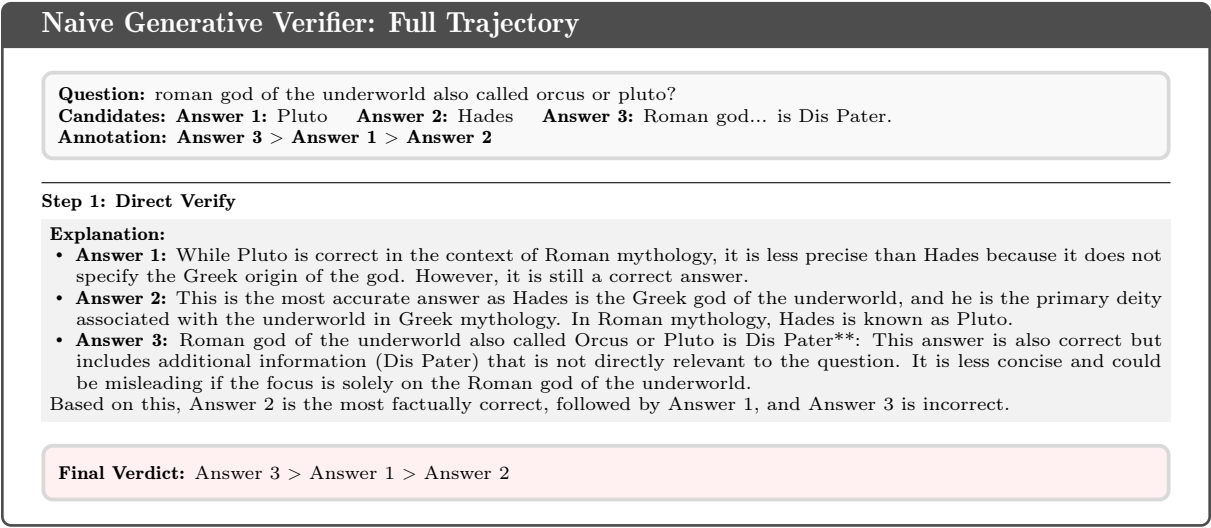


Figure 11: Full execution trace of the naive generative verification pipeline for general QA.

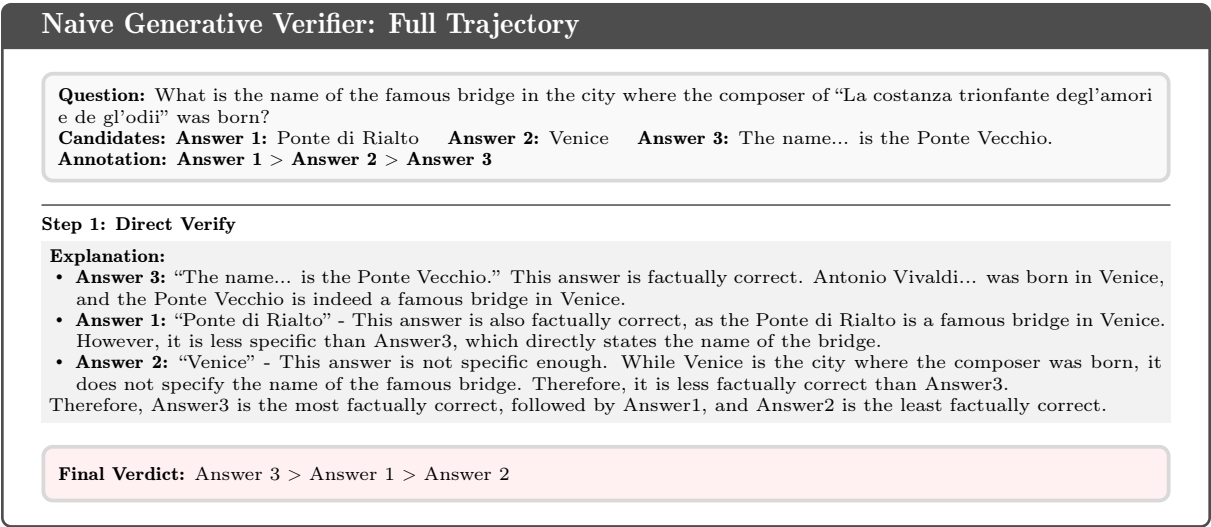


Figure 12: Full execution trace of the naive generative verification pipeline for multi-hop QA.

AEScorer: Full Trajectory

Question: roman god of the underworld also called orcus or pluto?
Candidates: **Answer 1:** Pluto **Answer 2:** Hades **Answer 3:** Roman god... is Dis Pater.
Annotation: Answer 3 > Answer 1 > Answer 2

Step 1: Agentic Search

Thought: I need to identify the Roman god of the underworld who is synonymous with Orcus or Pluto. I will start by searching local knowledge for these specific names.

Tool Call: search_local("roman god of the underworld orcus pluto")

Observations: [Source: Local Wikipedia]

- **Doc 2 ("Orcus"):** ...Orcus was a god of the underworld... In later tradition, he was conflated with Dis Pater.
- **Doc 3 ("Dis Pater"):** ...Dis Pater was a Roman god of the underworld... later equated with the chthonic deities Pluto (Hades) and Orcus.

Thought: Local results suggest Dis Pater, Pluto, and Orcus are conflated/synonymous. I will verify this with a web search to ensure these names refer to the same entity in this context.

Tool Call: search_web("roman god of the underworld orcus pluto")

Observations: [Source: Google Search]

- **Result 2:** In ancient Roman religion... Pluto also known as **Dis Pater** or Orcus, was the god of the dead...
- **Result 4: Dis Pater**, also known as Orcus, is the Roman god of the infernal regions, equivalent to the Greek Hades or Pluto.

Thought: The search results consistently confirm that Pluto, Orcus, and Dis Pater are synonymous names for the Roman god of the underworld. Hades is the Greek equivalent. **READY_FOR_ANSWERING**

Step 2: Context Compression

Compressed Context for Answer 1:

Useful Facts: Pluto is the Roman god of the underworld; Pluto is known as Dis Pater or Orcus; The names are interchangeable, but Dis Pater is often the formal Roman title.

Reasoning: 1. Search results consistently mention Pluto as the Roman god. 2. Information indicates Orcus and Dis Pater are conflated with Pluto. 3. Multiple sources confirm they are synonymous referring to the same god.

Compressed Context for Answer 2:

Useful Facts: Hades is the Greek god of the underworld; The Roman equivalent is Pluto/Dis Pater; Hades is not the correct name in a strictly Roman context.

Reasoning: 1. Orcus and Pluto are Roman gods. 2. Dis Pater is synonymous with them. 3. Hades is identified as the Greek god, not Roman, making this answer incorrect for the specific query.

Compressed Context for Answer 3:

Useful Facts: Dis Pater is explicitly stated to be the Roman god of the underworld; Dis Pater is synonymous with Orcus and Pluto; All three names refer to the same deity in Roman mythology.

Reasoning: 1. Search results confirm Orcus, Pluto, and Dis Pater are all Roman gods of the underworld. 2. Sources indicate Orcus and Pluto are synonymous. 3. Dis Pater is explicitly stated as equivalent to Pluto and Hades.

Step 3: Score Prediction

Answer 1: 0.4580

Answer 2: 0.2188

Answer 3: 0.5409

Figure 13: Full execution trace of agentic discriminative verification for general QA.

AEScorer: Full Trajectory

Question: What is the name of the famous bridge in the city where the composer of “La costanza trionfante degl’amori e de gl’odii” was born?
Candidates: **Answer 1:** Ponte di Rialto **Answer 2:** Venice **Answer 3:** The name... is the Ponte Vecchio.
Annotation: Answer 1 > Answer 2 > Answer 3

Step 1: Agentic Search

Thought: First, I need to find out in which city the composer of “La costanza trionfante degl’amori e de gl’odii” was born. Let’s start by searching for information about this opera and its composer.

Tool Call: search_local(“composer of La costanza trionfante degl’amori e de gl’odii”)

Observations: [Source: Local Wikipedia]

- **Doc 2 (“Griselda”):** ...Griselda is a dramma per musica... composed by Antonio Vivaldi.
- **Doc 3 (“Antonio Vivaldi”):** ...Also in 1716, Vivaldi wrote... “La costanza trionfante...” (RV 706).

Thought: The search result does not directly mention the composer... However, it mentions that Antonio Vivaldi... Let’s search for more information about it to confirm if it is the same opera and find the composer’s birthplace.

Tool Call: search_web(“composer of La costanza trionfante degli amori e degli odi”)

Observations: [Source: Google Search]

- **Result 1:** Also in 1716, Vivaldi wrote and produced... La costanza trionfante... (RV 706).
- **Result 2:** La costanza trionfante... is a dramma per musica by **Antonio Vivaldi**.

Thought: The information confirms that Antonio Vivaldi was the composer of “La costanza trionfante...”. Now, I need to find out where Antonio Vivaldi was born. Let’s search for his birthplace.

Tool Call: search_web(“birthplace of Antonio Vivaldi”)

Observations: [Source: Google Search]

- **Result 1:** Antonio Lucio Vivaldi was born on 4 March 1678 in **Venice**, then the capital...
- **Result 3:** The great composer... was indeed born in **Venice** and lived and worked here...

Thought: Based on the search results, Antonio Vivaldi was born in Venice, Italy. Now, I need to find the name of the famous bridge in Venice. Let’s search for a famous bridge in Venice.

Tool Call: search_web(“famous bridge in Venice”)

Observations: [Source: Google Search]

- **Result 1:** The **Rialto Bridge** (Ponte di Rialto)... is the oldest of the four bridges...
- **Result 8:** It is also the **most famous** in Venice thanks to its peculiar history...

Thought: Based on the search results, the Rialto Bridge is the most famous bridge in Venice. Therefore, the answer... is the Rialto Bridge. **READY_FOR_ANSWERING**

Step 2: Context Compression

Compressed Context for Answer 1:

Useful Facts: Antonio Vivaldi was the composer of “La costanza trionfante...”; Antonio Vivaldi was born in Venice, Italy; The Rialto Bridge (Ponte di Rialto) is the oldest and most famous bridge in Venice.

Reasoning: 1. The search history confirms that Antonio Vivaldi was the composer of “La costanza trionfante...”. 2. Further searches reveal that Antonio Vivaldi was born in Venice... 3. The search results also indicate that the Rialto Bridge... is the most famous bridge in Venice.

Compressed Context for Answer 2:

Useful Facts: Antonio Vivaldi was the composer of “La costanza trionfante...”; Antonio Vivaldi was born in Venice...; The Rialto Bridge is the oldest and most famous bridge in Venice.

Reasoning: 1. The search history confirmed that Antonio Vivaldi was the composer of “La costanza trionfante...”. 2. Further searches revealed that Antonio Vivaldi was born in Venice... 3. The search for famous bridges in Venice identified the Rialto Bridge as the most famous one.

Compressed Context for Answer 3:

Useful Facts: Antonio Vivaldi was the composer of “La costanza trionfante...”; Antonio Vivaldi was born in Venice...; The Rialto Bridge is the oldest and most famous bridge in Venice.

Reasoning: 1. The search history confirms that Antonio Vivaldi was the composer of “La costanza trionfante...”. 2. The search history also confirms that Antonio Vivaldi was born in Venice... 3. The search history provides information about famous bridges... specifically mentioning the Rialto Bridge...

Step 3: Score Prediction

Answer 1: 0.4448

Answer 2: 0.3479

Answer 3: 0.2119

Figure 14: Full execution trace of agentic discriminative verification for multi-hop QA.

Data Sample 1

Instruction: What nationality is Alice Delamar's father?
Reference: United States

Rank 1: American
Rank 2: I don't have information on Alice Delamar.
Rank 3: Alice Delamar's father's nationality is French.

Figure 15: A data sample constructed from 2Wiki (Ho et al., 2020) in GradedVeriBench.

Data Sample 2

Instruction: In the King's Speech, who played the person who was the King of England in 1950?
Reference: Colin Firth

Rank 1: In the the King's Speech, the actor who played the person who was King of England in 1950 is Colin Firth.
Rank 2: In the King's Speech, which is set in the 1930s, Colin Firth played the King of England, George VI. However, the question pertains to 1950, and by then, the King of England was still George VI, portrayed by Colin Firth in the film.
Rank 3: George VI

Figure 16: A data sample constructed from Musique (Trivedi et al., 2022) in GradedVeriBench.

PROMPT H.1: SYSTEM PROMPT FOR AGENTIC SEARCH

Your task is to determine the correct factual ranking of the provided answers. Based on the question and answers, think and identify what information you need and generate search queries using the available tools.

Question: {{question}}
{{answers_block}}

Leverage both tools ('search_local' for Wikipedia, 'search_web' for Google) and generate one search query per turn. If you have enough information, respond with 'READY_FOR_EVALUATION'.

PROMPT H.2: SYSTEM PROMPT FOR CONTEXT COMPRESSION

You are a fact-checking expert. Below is a history of search actions performed by an agent to gather information about a specific question and answer. Your task is to analyze this history and determine the factual correctness of the answer.

SEARCH HISTORY ###
{{search_history}}

VERIFICATION TASK

Based on the information from the search history, please verify the following:

Question: {{question}}
Answer to Verify: {{answer}}

ANALYSIS AND VERDICT

Based on your analysis, provide a structured response in the following format. Do not add any other text outside this structure.

Useful Facts: [List key facts from the search history relevant to the answer, separated by semicolons. Example: Fact1; Fact2; Fact3;]
Reasoning: [Provide a step-by-step reasoning for your verdict based on the useful facts.]

Final Verdict: [Your verdict: Correct, Incorrect, or Intermediate]

PROMPT H.3: SYSTEM PROMPT FOR NAIVE GENERATIVE VERIFICATION

You are an expert fact-checking assistant. Your task is to rank all the given answers from most to least factually correct based on your internal knowledge.

Question: `{{question}}`
`{{answers_block}}`

Please first provide your explanation, and then state your final verdict in the format: ‘**Final Verdict**: <verdict> AnswerX > AnswerY > AnswerZ </verdict>’. Example: ‘**Final Verdict**: <verdict>

Answer3 > Answer1 > Answer2 </verdict>’.

PROMPT H.4: SYSTEM PROMPT FOR AGENTIC GENERATIVE VERIFICATION

Based on the preceding conversation, your task is to rank all the given answers from most to least factually correct.

Question: `{{question}}`
`{{answers_block}}`

Please first provide your explanation, and then state your final verdict in the format: ‘**Final Verdict**: <verdict> AnswerX > AnswerY > AnswerZ </verdict>’. Example: ‘**Final Verdict**: <verdict>

Answer3 > Answer1 > Answer2 </verdict>’.

PROMPT H.5: SYSTEM PROMPT FOR AGENTIC GENERATIVE SCORING

Based on the preceding conversation, your task is to score the factuality of the given answer from a scale of 1-10.

Question: `{{question}}`
Answer: `{{answer}}`

Please first provide your explanation, and then state your final verdict in the format: ‘**Final Verdict**: <verdict> a score between 1-10 </verdict>’. Example: ‘**Final Verdict**: <verdict> 3

</verdict>’.