AKSH at CheckThat! 2026: A Three-Stage Multi-Agent Pipeline for Automated Fact-Checking Article Generation

Notebook for the CheckThat! Lab at CLEF 2026, Task 3

Syed Muhammad Kazim Raza[*,†], Faisal Alvi and Abdul Samad

[1]*Department of Computer Science, Dhanani School of Science and Engineering, Habib University, Karachi, Pakistan*

Abstract

This paper describes the system submitted by team AKSH to the CLEF 2026 CheckThat! Lab for Task 3 (Generating Full Fact-Checking Articles). Working with the WatClaimCheck dataset [1], we replace a single-prompt baseline with a structured three-stage multi-agent pipeline built on GPT-4o-mini, accessed via the OpenRouter API with no fine-tuning. Stage 1 (Nugget Extraction) distils noisy scraped evidence into concise, source-attributed factual nuggets. Stage 2 (Article Writing) employs real few-shot prompting from a dataset example to enforce journalistic structure; a bold title, analytical prose, and a mandatory verdict line, while explicitly suppressing hallucinated formatting artefacts. Stage 3 (Citation Verification) audits every factual sentence, binds inline citations exclusively to provided URLs, and enforces citation completeness across the article. On the official validation set, our pipeline substantially improved citation precision (0.2129 → 0.6125) and citation recall (0.2840 → 0.6687) over the baseline, along with modest evidence coverage gains (0.7628 → 0.8490), while entailment scores remained comparable (0.3713 → 0.3697). On the hidden test set of 1,158 articles, the official submission achieved a Mean Entailment Score of 0.329, Mean Citation Precision of 0.301, Mean Citation Recall of 0.318, Mean Evidence Coverage of 0.564, and a composite Mean Score of 0.378. We provide a detailed analysis of the improvements, the generalisation gap between validation and hidden-test performance, and the engineering challenges encountered throughout development.

Keywords

fact-checking article generation, RAG, citation verification, few-shot prompting, WatClaimCheck, GPT-4o-mini

1. Introduction

Automated fact-checking has attracted sustained research attention as a countermeasure to the accelerating spread of online misinformation. While early work focused primarily on binary veracity classification [2, 3], the practical requirements of professional fact-checkers demand considerably more: a complete, structured article that situates a claim within its evidentiary context, assembles supporting or refuting evidence with precise inline attribution, and delivers a clearly reasoned verdict. Generating such articles automatically is substantially harder than classification because it requires the system to simultaneously retrieve relevant evidence, maintain factual grounding, suppress hallucination, and produce coherent long-form text [4].

The CLEF 2026 CheckThat! Lab [5] formalises this challenge as Task 3: *Generating Full Fact-Checking Articles* [6]. Given a real-world claim together with a set of pre-retrieved evidence documents, participating systems must generate a complete, publication-quality fact-checking article with inline citations grounded exclusively in the provided evidence. The task is evaluated on the WatClaimCheck dataset [1], which pairs newsworthy claims with human-authored fact-check articles and their underlying reference documents, the same resource used in the construction of the ExClaim benchmark for justification generation [7].

The central challenge of this task is not generation fluency per se, but *verifiable generation*: producing text in which every factual assertion is traceable to a cited source and the citation accurately reflects

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

[*]Corresponding author.

[†]These authors contributed equally.

sr09142@st.habib.edu.pk (S. M. K. Raza); faisal.alvi@sse.habib.edu.pk (F. Alvi); abdul.samad@sse.habib.edu.pk (A. Samad)

the evidence it purports to support [8, 9, 10]. Large language models (LLMs) readily produce plausible-sounding prose but are prone to citation hallucination, inventing URLs, attributing claims to sources that do not support them, or omitting citations for verifiable facts altogether. These failure modes are particularly damaging in a fact-checking context, where source attribution is the primary mechanism of accountability.

Our system, submitted under the team name **AKSH**, addresses this problem through a three-stage pipeline in which generation, evidence processing, and citation verification are modularised into separate agents, each with a precisely scoped prompt. This design is motivated by the observation that decomposing complex generation tasks into specialised sub-problems consistently outperforms monolithic prompting [11], a principle directly instantiated in our nugget extraction, article writing, and citation verification stages. We further incorporate real few-shot prompting from an actual dataset example to anchor the model's output style, drawing on the finding in prior work that few-shot in-context learning substantially improves both structural compliance and factual grounding in generation tasks [7, 11].

The remainder of this paper is structured as follows. Section 2 formally defines Task 3 and the evaluation protocol. Section 3 situates our approach within prior work on fact-checking generation, verifiable text generation, and RAG-based pipelines. Section 4 describes our full three-stage methodology. Section 5 presents results on the validation and hidden test sets. Section 6 provides an analysis of performance patterns and failure modes. Section 7 concludes with directions for future work.

2. Task Description

Task 3 of the CLEF 2026 CheckThat! Lab is titled *Generating Full Fact-Checking Articles* [6]. It operationalises the article-generation component of the professional fact-checking workflow: given structured inputs about a claim, a system must produce a complete journalistic article that explains the verdict with evidentiary support and precise inline citations.

2.1. Problem Formulation

Each instance provides the following inputs:

- **Claim**: a naturally occurring, newsworthy statement to be fact-checked (e.g., *"Gun manufacturers are the only industry exempt from lawsuits in the US"*).
- **Claimant**: the individual or organisation who made the claim, when known.
- **Original Rating**: the veracity label assigned by the originating fact-checking outlet.
- **Evidence**: a dictionary mapping URLs to scraped article content, constituting the set of reference documents the system may draw upon.

Systems must produce a single output per instance:

- **Fact-Checking Article**: a complete written article containing a headline, analytical body with inline source citations, and a verdict conclusion. Citations must follow the format `(source: <url>)` and reference only URLs provided in the evidence inputs.

2.2. Dataset

The task uses the WatClaimCheck dataset [1], which provides real-world newsworthy claims alongside their fact-check articles and the reference documents cited within those articles. The dataset was previously used to construct the ExClaim benchmark for few-shot justification generation [7]. Table 1 summarises the split sizes relevant to our experiments.

Table 1
WatClaimCheck split sizes as used in Task 3.

Split	Instances
Validation	200
Hidden Test	1,158

2.3. Evaluation Metrics

The official evaluation uses four complementary metrics, each targeting a distinct quality dimension:

Entailment Score: a bidirectional NLI-based score computed using `roberta-large-mnli`, applied in a sliding-window fashion (chunk size 3, overlap 0) over the generated article and the gold reference. Bidirectional entailment encourages both recall of reference content and precision of generated claims.

Citation Precision: assessed via `llama3.2:1b` as a local entailment judge. For each inline citation present in the generated article, the evaluator checks whether the cited source actually supports the sentence it is attached to. Precision rewards well-grounded, non-hallucinated citations.

Citation Recall: the complement metric, measuring whether evidence present in the provided sources is cited at all in the generated article. Recall penalises omission of relevant evidence.

Evidence Coverage: a surface-level metric that checks whether the generated article incorporates content from across the full set of provided evidence URLs, rewarding breadth of evidence utilisation.

A composite **Mean Score** averages the four metrics into a single leaderboard figure.

3. Related Work

3.1. Fact-Checking Article and Verdict Generation

Automated generation of fact-checking explanations has evolved from attention-based token highlighting [12] toward abstractive generation over retrieved evidence. Early approaches treated the problem as summarisation of existing fact-check articles, using extractive models [13] or encoder-decoder abstractive systems [14]. A fundamental limitation of this paradigm, one that Task 3 directly targets, is that gold fact-check articles are unavailable for new claims at inference time. Zeng and Gao [7] formalised this constraint through JustiLM, a few-shot retrieval-augmented generation model trained on WatClaimCheck that uses fact-check articles only during training and generates justifications from retrieved evidence at inference time. Our work operates in an analogous regime but is entirely inference-time, relying on prompt engineering rather than fine-tuning, a design forced by API-only access constraints.

Russo et al. [11] systematically benchmarked RAG-based verdict generation across multiple retriever configurations, LLM sizes, and generation setups (zero-shot, one-shot, fine-tuned) on the FullFact and VerMouth datasets. Their findings inform two specific choices in our pipeline: the adoption of one-shot prompting (which they find consistently competitive with zero-shot on informativeness) and explicit structural prompting to enforce journalistic formatting, which reduces content drift in generated verdicts.

3.2. Verifiable Text Generation and Citation Grounding

Verifiable text generation, that is producing long-form outputs annotated with citations to supporting documents, has emerged as a distinct subfield motivated by LLM hallucination. Sun et al. [8] propose the VTG framework, which decomposes generation into sequential claim generation and citation assignment steps, using a two-tier NLI verifier to assess whether cited documents entail each generated claim. The separation of generation from attribution in VTG directly motivates our modular three-stage design. Gao et al. [9] introduced the ALCE benchmark and demonstrated that prompted LLMs can generate text with inline citations provided sufficient evidence context, but that citation recall degrades sharply when the evidence is long or noisy, precisely the condition posed by raw scraped web content in

WatClaimCheck. Our nugget extraction stage is designed to address this failure mode by pre-distilling evidence before generation.

3.3. Retrieval-Augmented Generation for Fact-Checking

Retrieval-augmented generation has become standard in knowledge-intensive NLP tasks [15], and its application to fact-checking is well established. In the context of the CheckThat! Lab, prior systems have used BM25 [16], cross-encoders, and dense retrievers to identify relevant evidence before classification or generation. Russo et al. [11] find that LLM-based embedding retrievers consistently outperform sparse and dense alternatives when the knowledge base is heterogeneous, though silver knowledge bases analogous to our noisy scraped evidence remain challenging for all retrieval configurations. Since Task 3 provides the evidence set directly without requiring retrieval, our system focuses its engineering effort on evidence cleaning and citation binding rather than retrieval.

3.4. Few-Shot Prompting for Generation Quality

Few-shot in-context learning has been shown to substantially improve both structural compliance and factual fidelity in generation tasks. Zeng and Gao [7] demonstrate that even a small number of high-quality demonstrations significantly improves justification coherence for WatClaimCheck claims compared to zero-shot generation. In our system, we use a real dataset example as a style reference rather than a hand-crafted demonstration, which grounds the model's output format in the actual distribution of the target articles. This approach avoids the risk of the model over-fitting to a manually constructed demonstration that may not reflect the stylistic diversity of the full dataset.

4. Methodology

Our system is a three-stage, prompt-only multi-agent pipeline built on GPT-4o-mini, accessed via the OpenRouter API. No fine-tuning, external retrieval augmentation, or additional training data beyond the task-provided evidence was used. All inference is performed with `temperature=0.0` and a fixed random seed for reproducibility. Concurrency is capped at 3 simultaneous API requests, each with a 45-second timeout. The full pipeline is illustrated schematically in Figure 4.

4.1. Baseline System

The provided baseline concatenates raw evidence content directly into a single prompt, instructs the model to write a fact-checking article with inline citations, and performs generation in one pass with no preprocessing or post-hoc verification. This approach suffers from three structural weaknesses. First, raw scraped web content contains substantial boilerplate, including navigation menus, cookie notices, subscription prompts, and social media sharing buttons, all of which dilute the evidence signal and mislead the citation model. Second, without explicit structural constraints, LLMs tend to produce inconsistent output formats: bullet points instead of prose, omitted verdict lines, or improperly formatted citation markers. Third, without citation verification, the model freely cites hallucinated URLs or attributes claims to sources that do not support them. Table 2 summarises baseline performance on the validation set.

4.2. Evidence Cleaning

A critical preprocessing decision that applies uniformly across all stages is evidence cleaning. Raw evidence in WatClaimCheck is extracted from the live web and contains substantial noise. We implement a rule-based cleaning function `clean_evidence()` that processes evidence content through the following filters:

- Lines shorter than 50 characters are discarded as likely UI fragments.

- Lines beginning with Markdown-style list tokens (`*`, `#`, `|`, `>`) or bare URLs are removed.
- A compiled regular expression pattern strips lines that match common boilerplate prefixes: navigation labels (*skip to*, *home*, *about*), call-to-action phrases (*subscribe*, *donate*, *sign up*), and legal footer text (*terms of service*, *privacy policy*, *copyright*).
- Lines with more than two asterisk or three pipe characters are removed as table or formatting artefacts.

Similarly, a `clean_reference_article()` function applies the same logic to the gold reference articles used as the few-shot style example, adding supplementary patterns specific to known fact-checking outlet footers (e.g., `FactCheck.org`, `Annenberg`, `Previous Story`).

After cleaning, evidence content is truncated to 4,000 characters per source to remain within context-window limits of GPT-4o-mini under high-concurrency operation.

4.3. Stage 1: Nugget Extraction Agent

The first stage converts the cleaned, per-URL evidence texts into structured factual nuggets. The model receives a system message instructing it to act as a fact-checking research assistant and to extract at least one key factual sentence per source, with strict attribution preserved. The output format is a JSON array of objects, each with a `url` field and a `nugget` field:

```
[{"url": "<url>", "nugget": "<key fact sentence>"}, ...]
```

The prompt explicitly prohibits skipping any source, ensuring coverage across the full evidence set. JSON output is requested without markdown fences or explanatory preamble. If JSON parsing fails, which can occur when the model emits a malformed response, a fallback assigns each cleaned evidence text as its own nugget, preserving pipeline continuity.

The design rationale for this stage is twofold. First, by compressing evidence into nuggets before generation, the Stage 2 prompt receives a substantially shorter, higher-density input, reducing the probability that key evidence is ignored due to context-window dilution. Second, pre-extracting nuggets with source attribution makes citation binding in Stage 3 more tractable: the verifier can check whether a citation matches a specific factual claim rather than navigating an entire scraped article.

4.4. Stage 2: Article Writing Agent

The second stage generates the full fact-checking article from the extracted nuggets. Two design choices distinguish this stage from the baseline.

Real few-shot style reference. Rather than prompting zero-shot or constructing an artificial demonstration, we use the first example from the test set as a live style reference. The associated gold reference article is cleaned with `clean_reference_article()`, and a style-reference block is prepended to the user prompt, explicitly labelling the example as a structural guide and instructing the model to match its citation format and journalistic tone without copying its facts or URLs. This approach grounds the model's output in the actual target distribution rather than in a hand-authored example that may not reflect the range of claim topics.

Explicit structural prompt engineering. The system prompt `ARTICLE_SYSTEM_PROMPT` enforces a strict three-part output format:

1. A bold title on the first line, written as plain bold markdown (`**Fact-Check: ...**`). The model is explicitly forbidden from prepending a `Headline:` label.
2. An **Analysis** section with bold subheaders for each country, person, or sub-topic examined. Subheaders must be on their own lines. Bullet points and numbered lists are strictly prohibited anywhere in the article body.

3. A **Conclusion** section ending with a `**Verdict: <rating>**` line.

Citation format is mandated throughout: every factual sentence must end with `(source: <url>)`, and multi-source citations must use `(source: <url1>; <url2>)`. Only URLs from the provided evidence nuggets may appear in citations. The human-turn template `ARTICLE_HUMAN_TEMPLATE` repeats structural reminders in its closing instruction to reduce formatting drift across long contexts.

The article prompt is constructed with the following inputs:

```
CLAIM: {claim}
CLAIMANT: {claimant}   [omitted if unknown]
VERDICT: {original_rating}
KEY EVIDENCE NUGGETS (cite each inline):
- {nugget} (source: {url}) × N
```

Post-processing. A `clean_final_article()` function applies four targeted corrections to the generated text: stripping any metadata header block that the model emits before the title (`Claim:`, `Claimant:`, `Original rating:`); removing hallucinated boilerplate footers (e.g., `FactCheck.org®`, editor's notes); removing erroneous `Headline:` labels from the title line; and collapsing sequences of three or more blank lines into a single blank line.

4.5. Stage 3: Citation Verification Agent

The third stage audits the draft article produced by Stage 2. The system prompt `VERIFY_SYSTEM_PROMPT` instructs the model to act as a citation editor, adding or correcting inline citations according to the following rules:

- Every factual sentence must carry an inline citation in `(source: <url>)` format.
- Only URLs from the valid sources list provided in the prompt are permitted; hallucinated citations must be removed.
- Factual content, structure, and markdown formatting must be preserved verbatim; the verifier may only modify citation markers.
- The article must begin directly with the bold title; no preamble or metadata header should be present.

The verifier receives the list of valid source URLs separately from the draft article text, enabling it to cross-check each citation without needing to re-read the full evidence corpus. The output of Stage 3 is the final article submitted for evaluation.

Source credibility guidance. In addition to citation correctness, the system prompt instructs the model to prioritise citations from high-authority sources, such as official government agencies, the WHO, and established news organisations, when multiple sources are available for a given claim. Social media posts, unverified blogs, and sensationalist websites are identified as weak evidence and treated with reduced weight. This guidance was introduced after observing that the Stage 2 model sometimes allowed low-credibility sources to dominate the narrative, harming entailment alignment with the gold reference.

4.6. Inference and Batching

All three stages are implemented as asynchronous LangChain `ChatOpenAI` calls batched in groups of up to 100 instances. Intermediate outputs are flushed to disk after each batch, ensuring that a partial run can be resumed without reprocessing completed instances. The final JSON output maps each instance `id` to its generated `factchecking_article`.

4.7. Development Systems

We developed and evaluated six systems in total, each building on the previous one by adding or changing one component at a time. This incremental design made it possible to trace the effect of each individual change on the evaluation metrics. All six systems were evaluated on the same 20-example subset drawn from the validation set.

The **Baseline** system is the task-provided single-call pipeline. It feeds raw scraped evidence directly into one prompt and generates the article in a single pass, with no preprocessing, no structural constraints, and no citation verification.

The **Initial Approach** keeps the same single-call structure but rewrites the prompt with strict citation rules. Every factual sentence is required to end with an inline citation, only the exact provided URLs may be used, and every URL must appear at least once in the article. The human prompt also lists all valid URLs explicitly before the evidence text and structures the output into five ordered steps: verdict, headline, analysis, citation coverage, and conclusion.

The **Nugget** system introduces a three-stage pipeline for the first time. In Stage 1, a separate LLM call processes each evidence document and extracts at least one key factual sentence per source, outputting a JSON list of url-nugget pairs. Stage 2 generates the article from these nuggets rather than from raw evidence. Stage 3 runs a citation verification pass over the draft article, checking that every factual sentence carries a valid inline citation from the provided URL list. Temperature is also lowered from 0.6 to 0.0 at this point.

The **Synthetic Few-Shot** system keeps the same three-stage structure but adds a hand-written example article to the Stage 2 system prompt. This example is a made-up fact-check about a bleach and COVID claim, written to demonstrate the target structure and citation style. It teaches the model what a complete fact-checking article should look like without revealing any real dataset facts.

The **Combined** system takes a different direction. It drops nugget extraction entirely and goes back to feeding raw evidence directly into the article generation prompt, as the baseline does. It combines the strict citation rules from the Initial Approach with the citation verification stage from the Nugget system, and sets temperature back to 0.6. The intention was to check whether the gains from citation rules and verification alone, without evidence compression, could match or exceed the nugget-based pipeline.

The **Real Few-Shot** system is the final submitted pipeline. It builds on the Nugget system by replacing the synthetic example with a real article from the WatClaimCheck dataset. The first example in the validation set is excluded from evaluation and its cleaned reference article is used as the style guide in Stage 2. Evidence cleaning with `clean_evidence()` is also introduced at this stage, removing boilerplate lines and navigation fragments before truncating each document to 4,000 characters.

5. Results

We report performance on two evaluation surfaces: the validation set (Section 5.1), used for iterative development and ablation, and the official hidden test set (Section 5.2), evaluated by the task organisers after submission.

A critical caveat governs the comparison between these two figures. During development, iterative testing was performed on small subsets of approximately 20 samples at a time, imposed by API credit and runtime constraints. Qualitative improvements observed at this scale informed prompt design decisions but could not be statistically validated on the full dataset. The official hidden test set comprises 1,158 articles, roughly 58 times the typical development batch, and exposes the system to the full distributional range of WatClaimCheck claims and evidence quality. Readers should therefore treat the validation and hidden-test figures as measuring related but distinct phenomena: the former reflects development-time behaviour under controlled conditions, while the latter reflects true generalisation performance.

5.1. Validation-Set Performance

Table 2 reports all six systems on the four evaluation metrics on the validation set. The systems are ordered by development sequence to show how each change affected the scores.

Table 2
Validation-set performance: all six systems.

System	Entailment	Cit. Prec.	Cit. Rec.	Evid. Cov.
Baseline	0.3713	0.2129	0.2840	0.7628
Initial Approach	0.3037	0.4270	0.4805	0.7283
Nugget	0.2887	0.5150	0.5400	0.4790
Synthetic Few-Shot	0.3036	0.5433	0.5433	0.7711
Combined	0.3001	0.4090	0.4779	0.7670
Real Few-Shot	**0.3697**	**0.6125**	**0.6687**	**0.8490**

The most striking gains are in citation quality. Moving from the Baseline to the Initial Approach already brings citation precision from 0.2129 to 0.4270 and citation recall from 0.2840 to 0.4805, which shows that stricter prompt rules alone make a meaningful difference even without any structural changes to the pipeline. Adding nugget extraction in the Nugget system pushes citation precision further to 0.5150 and recall to 0.5400, but evidence coverage drops sharply from 0.7283 to 0.4790. This drop happens because compressing evidence into nuggets before generation causes the model to lose track of some source URLs during article writing, even when the nugget prompt explicitly requires at least one extraction per source.

The Synthetic Few-Shot system partially recovers evidence coverage to 0.7711 by anchoring the model's output format to a concrete example, and also nudges citation precision upward to 0.5433. The Combined system, which drops nugget extraction and returns to raw evidence, confirms that evidence compression is what hurts coverage: going back to raw evidence brings coverage back to 0.7670. However, without nugget extraction, citation precision falls back to 0.4090, well below the nugget-based systems.

The Real Few-Shot system resolves this tension. Evidence cleaning with `clean_evidence()` removes boilerplate from the raw scraped text before nugget extraction, so the nuggets are built from cleaner input. Using a real dataset article as the few-shot example rather than a synthetic one gives the model a more accurate style target. Together these two changes bring citation precision to 0.6125, citation recall to 0.6687, and evidence coverage to 0.8490, the best scores across all four metrics. Entailment also recovers to 0.3697, nearly matching the baseline figure of 0.3713 after having dropped across the intermediate systems.

The flat entailment score across most systems is informative. Entailment measures semantic consistency between the generated article and the gold reference. The baseline already produces reasonably fluent prose that captures the general thrust of a claim's veracity, and adding structural and citation enforcement does not substantially alter the propositional content of the generated text. This suggests that the primary bottleneck for entailment improvement is not citation quality but rather the depth of reasoning and factual precision of the generated content, qualities that prompt engineering alone, without fine-tuning, is ill-suited to target.

5.2. Hidden Test-Set Performance

Table 3 reports the official results from the hidden test set, released on 12 May 2026.

Across all four metrics, hidden-test performance is substantially lower than the corresponding validation figures, with citation precision declining from 0.6125 to 0.301, citation recall from 0.6687 to 0.318, evidence coverage from 0.8490 to 0.564, and entailment from 0.3697 to 0.329. We analyse the sources of this degradation in Section 6.

Table 3
Official hidden test-set results (1,158 articles, released 12 May 2026).

Metric	Score
Mean Entailment Score	0.329
Mean Citation Precision	0.301
Mean Citation Recall	0.318
Mean Evidence Coverage	0.564
Mean Score (composite)	**0.378**

6. Analysis

6.1. Citation Quality Improvements and Their Limits

The substantial validation-set gains in citation precision and recall confirm that the combination of nugget extraction and citation verification is effective at the unit level. Stage 1 ensures that each source contributes at least one explicitly attributed factual claim to the article prompt. Stage 3 verifies that these attributions survive into the final output and binds every factual sentence to a URL drawn from the provided evidence. Together, these two stages eliminate the two principal failure modes of the baseline: citation omission (addressed by recall-oriented Stage 1) and citation hallucination (addressed by precision-oriented Stage 3).

The hidden-test degradation in citation scores, with precision dropping from 0.6125 to 0.301, indicates that this pipeline is less robust at scale than the validation figures suggest. The most likely cause is evidence quality variance: WatClaimCheck evidence documents span a wide range of source types and extraction quality, and the cleaning heuristics designed on a small development sample do not generalise perfectly to the full distributional range. When cleaned evidence is sparse or incoherent, Stage 1 nuggets are incomplete, Stage 2 generates assertions without strong evidentiary anchors, and Stage 3 cannot bind citations that were never present in the nuggets. The pipeline degrades gracefully rather than catastrophically, but the cumulative effect is substantial.

6.2. Entailment: Structural versus Semantic Alignment

The near-zero change in entailment scores across the transition from baseline to our system (validation: $0.3713 \rightarrow 0.3697$; hidden test: $\sim$0.329) reflects a fundamental architectural limitation. The entailment evaluator measures whether the generated article makes claims that are semantically consistent with the gold reference. Improving citation correctness does not directly improve semantic consistency unless the cited content aligns with the factual propositions in the gold article.

There are two mechanisms through which entailment could be improved beyond what prompt engineering achieves. First, supervised fine-tuning on WatClaimCheck training examples would allow the model to learn the specific reasoning style and level of factual specificity expected in the gold references, a form of target-distribution adaptation that zero-shot and few-shot prompting cannot replicate. JustiLM [7] demonstrates that fine-tuning a RAG model on this dataset yields consistent ROUGE and SummaCC improvements over all prompted baselines, even when the LLM is substantially larger. Second, a dedicated entailment-aware generation loop, akin to the VTG framework's two-tier verifier [8], could iteratively refine generated claims by checking them against the evidence before finalising the article. Neither approach was feasible under the API-only, credit-constrained experimental setting.

6.3. Generalisation Gap

The validation-to-hidden-test degradation is substantial and warrants careful interpretation. Several factors likely contribute.

Scale mismatch. Development was conducted on batches of approximately 20 samples, chosen for API cost and iteration speed. Prompt design decisions were validated qualitatively on this small sample. The hidden test set of 1,158 instances exposes the system to a far wider range of claim topics, evidence quality, and evidence lengths. Heuristics that worked well on the development sample, such as the specific boilerplate patterns in `clean_evidence()`, may not transfer to the full data distribution.

Evidence quality variance. WatClaimCheck evidence is scraped from live web sources at the time of dataset construction. The quality of scraped content varies enormously: some sources provide clean, dense factual text, while others consist primarily of navigation chrome and advertising content. At scale, the proportion of low-quality evidence articles is likely higher than in the small development subset, directly degrading nugget extraction and downstream citation quality.

Prompt engineering limits. All structural and formatting constraints in our system are enforced exclusively through natural language instructions. GPT-4o-mini does not uniformly comply with these instructions across all claim types and evidence configurations. Formatting failures, such as bullet points where prose was requested, missing verdict lines, or improperly labelled titles, occur at a non-trivial rate at scale even when they are rare in small-sample testing. Without a programmatic output parser or constrained decoding, these failures propagate into the submitted articles and affect all downstream metrics.

Source credibility sensitivity. The source credibility guidance in Stage 3 introduces an additional prompt instruction that can cause the verifier to restructure citation patterns in ways that reduce rather than increase coverage when low-credibility sources carry significant evidentiary weight for a given claim. In the small development sample, the guidance improved qualitative output; at scale, it introduces variance that is difficult to control without more precise authority signals.

6.4. Engineering Challenges

Several practical challenges shaped the final system design in ways not fully reflected by metric comparisons.

Model accessibility constraints. GPT-5-mini was the originally intended backbone but was financially infeasible under available API credits. We substituted GPT-4o-mini, which is substantially cheaper but demonstrates less consistent compliance with complex multi-rule prompts, particularly the simultaneous enforcement of prose structure, citation format, and source credibility ordering. This tradeoff between affordability and reasoning quality is a recurring constraint in shared-task settings and directly limits the ceiling of prompt-only approaches.

Output format instability. Despite extensive prompt engineering, the model intermittently produced bullet points, omitted subheaders, or emitted a `Headline:` label before the bold title. The `clean_final_article()` post-processing function addresses the most common failure modes, but it operates on surface patterns and cannot recover articles that have structurally diverged from the required format. A constrained decoding approach, enforcing format compliance at the token level, would eliminate this failure class entirely but was not available through the OpenRouter API.

Iterative testing bottleneck. The 20-sample iterative testing regime, though necessary given resource constraints, meant that prompt improvements were validated on a sample far too small to detect generalisation failures. Improvements that appeared reliable in qualitative inspection over 20 examples sometimes did not replicate at scale. Future work should prioritise a larger validation subset, even 100–200 samples, for prompt iteration before final submission.

7. Conclusion

We presented the AKSH system for CLEF 2026 CheckThat! Task 3: Generating Full Fact-Checking Articles. Our three-stage multi-agent pipeline, comprising Nugget Extraction, Article Writing, and Citation Verification, replaces the monolithic single-prompt baseline with a modular design in which each stage performs a precisely scoped subtask. Operating entirely through prompt engineering on GPT-4o-mini, with no fine-tuning or external retrieval, the pipeline achieved substantial improvements in citation precision (+0.40 absolute) and citation recall (+0.38 absolute) on the validation set relative to the baseline, alongside a modest evidence coverage gain (+0.09 absolute). Entailment scores remained essentially unchanged, confirming that citation grounding and semantic factual alignment are separable quality dimensions that require distinct interventions.

On the official hidden test set of 1,158 articles, the system achieved a composite Mean Score of 0.378, with performance substantially below validation figures across all metrics. This generalisation gap reflects a combination of scale mismatch between development-time testing and final evaluation, evidence quality variance in the full dataset, and the inherent instability of prompt-only structural enforcement at scale.

The core lessons from this work are threefold. First, modular multi-agent decomposition substantially improves citation quality over monolithic prompting: separating nugget extraction from article generation from citation verification allows each agent to focus on a single well-defined objective. Second, few-shot prompting from real dataset examples is preferable to zero-shot or hand-crafted demonstrations: anchoring the model's output format to an actual target-distribution article reduces structural drift without requiring manual demonstration design. Third, the ceiling of prompt-only approaches under resource-constrained API access is real and meaningful: improvements in entailment and full-dataset generalisation will likely require supervised fine-tuning, retrieval-augmented evidence grounding, or programmatic output validation that is not achievable through natural language instructions alone.

Future Work. We identify four concrete directions for improvement. *(i) Supervised fine-tuning:* fine-tuning on WatClaimCheck training examples using the distillation techniques of JustiLM [7] would provide direct target-distribution adaptation for both the retriever and the generation model. *(ii) Entailment-aware generation loop:* incorporating a two-tier NLI verifier akin to VTG [8] to iteratively check and regenerate factually inconsistent claims before finalising the article. *(iii) Programmatic format enforcement:* replacing prompt-only structural constraints with a constrained decoding or post-hoc parse-and-repair approach to eliminate format failures at scale. *(iv) Authority-weighted evidence ranking:* adding a dedicated source credibility scoring step before nugget extraction, drawing on authority signals such as domain reputation or publisher verification, to reduce the influence of low-quality scraped content on downstream generation.

Acknowledgments

We thank the CLEF 2026 CheckThat! Lab organisers for designing the tasks, providing the datasets, and operating the CodaLab evaluation infrastructure. We would like to thank the faculty and teaching staff of CS/CE 335/466 (Introduction to Large Language Models) at Habib University for their guidance throughout this project.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) to assist with drafting and proofreading of this manuscript. GPT-4o-mini (OpenAI, accessed via OpenRouter) was used as the generation backbone within the experimental system described in Section 4. After using these tools,

the authors reviewed and edited all content as needed and take full responsibility for the publication's content.

References

[1] K. Khan, R. Wang, P. Poupart, WatClaimCheck: A new dataset for claim entailment and inference, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 1293–1304. doi:10.18653/v1/2022.acl-long.92.

[2] A. Vlachos, S. Riedel, Fact checking: Task definition and dataset construction, in: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, Association for Computational Linguistics, Baltimore, MD, USA, 2014, pp. 18–22. doi:10.3115/v1/W14-2508.

[3] W. Y. Wang, "Liar, Liar pants on Fire": A new benchmark dataset for fake news detection, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, 2017, pp. 422–426. doi:10.18653/v1/P17-2067.

[4] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206. doi:10.1162/tacl_a_00454.

[5] E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CLEF 2026, Jena, Germany, 2026.

[6] D. Sahnan, T. Chakraborty, P. Nakov, Overview of the CLEF-2026 CheckThat! lab task 3 on generating full fact-checking articles, in: [5], 2026.

[7] F. Zeng, W. Gao, JustiLM: Few-shot justification generation for explainable fact-checking of real-world claims, Transactions of the Association for Computational Linguistics 12 (2024) 334–354. doi:10.1162/tacl_a_00649.

[8] H. Sun, H. Cai, B. Wang, Y. Hou, X. Wei, S. Wang, Y. Zhang, D. Yin, Towards verifiable text generation with evolving memory and self-reflection, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2024, pp. 8211–8227.

[9] T. Gao, H. Yen, J. Yu, D. Chen, Enabling large language models to generate text with citations, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2023, pp. 6465–6488. doi:10.18653/v1/2023.emnlp-main.398.

[10] H. Rashkin, V. Nikolaev, M. Lamm, L. Aroyo, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, D. Reitter, Measuring attribution in natural language generation models, Computational Linguistics 49 (2023) 777–840. doi:10.1162/coli_a_00486.

[11] D. Russo, S. Menini, J. Staiano, M. Guerini, Face the facts! evaluating RAG-based pipelines for professional fact-checking, arXiv preprint arXiv:2412.15189 (2024).

[12] K. Popat, S. Mukherjee, A. Yates, G. Weikum, DeClarE: Debunking fake news and false claims using evidence-aware deep learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2018, pp. 22–32. doi:10.18653/v1/D18-1003.

[13] P. Atanasova, J. G. Simonsen, C. Lioma, I. Augenstein, Generating fact checking explanations, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 7352–7364. doi:10.18653/v1/2020.acl-main.656.

[14] N. Kotonya, F. Toni, Explainable automated fact-checking for public health claims, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2020, pp. 7740–7754. doi:10.18653/v1/2020.emnlp-main.623.

[15] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih,

T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 9459–9474.

[16] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-3, in: Proceedings of the Third Text REtrieval Conference (TREC-3), volume 500-225, National Institute of Standards and Technology (NIST), 1994, pp. 109–126.