Overview of the CLEF-2026 CheckThat! Lab Task 3 on Generating Full Fact-Checking Articles

Dhruv Sahnan[1], Tanmoy Chakraborty[2] and Preslav Nakov[1]

[1]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi
[2]Indian Institute of Technology Delhi, India

Abstract

The CLEF CheckThat! lab has been pivotal in advancing research on automated fact-checking by organizing shared tasks that target the various sub-tasks of the professional fact-checking workflow. Yet, one important and time-consuming sub-task of this workflow has remained largely unexplored: the generation of full fact-checking articles. In professional fact-checking, the article is the final public-facing artifact that guides readers toward a comprehensive understanding of the claim, its veracity, and the surrounding narrative through relevant evidence. At the CLEF 2026 CheckThat! lab, we introduce this task for the first time, focusing on the automatic generation of full fact-checking articles with inline citations. In this paper, we provide an overview of the shared task, including details on the datasets used, our evaluation methodology, the approaches proposed by the participating systems, and the main findings. The task attracted submissions from 10 teams, with one additional team submitting after the official competition deadline. We evaluate submissions using a dual-focus evaluation strategy that combines automatic measures with LLM-as-a-judge evaluations for assessing information coverage and evidence grounding as well as assessing the writing quality of the articles in terms of clarity, coherence, and comprehensibility as expected by professional standards of writing. By introducing this task, we aim to stimulate further research on this relatively new and important direction to support the development of systems that can help expedite professional fact-checking workflows to match the scale of global mis- and disinformation. The code and datasets are publicly available at this URL.

Keywords

Fact-Checking, Misinformation, Fact-Checking Article Generation, Long-Form Generation, Reasoning

1. Introduction

Digital media and social media platforms have revolutionized global communication, transforming the way information is produced, shared, and consumed. These platforms provide users with easy access to large and diverse audiences, enable rapid dissemination of breaking news, and allow users to participate in public discourse with limited gate-keeping and often (pseudo)anonymously. At the same time, these same advantages make online platforms a breeding ground for unverified and dubious claims. Such claims can quickly become viral, particularly during breaking-news events like elections [1, 2], or public-health crises [3], potentially causing harm to the masses.

In response, professional fact-checking has become increasingly important for maintaining information integrity, with several fact-checking initiatives emerging in recent years, such as Newtral, Full Fact, and Pagella Politica, among others [1]. However, manual fact-checking is meticulous and is therefore time-consuming, making it difficult for professional fact-checkers to match the scale of global mis- and disinformation. This has motivated substantial research on automating different stages of the fact-checking pipeline [4]. Existing work has typically decomposed automatic fact-checking into several sub-tasks, namely, (i) check-worthy claim detection, (ii) previously verified claim retrieval, (iii) evidence retrieval, (iv) claim verification, and (v) explanation generation [5, 6, 7]. The CLEF CheckThat! lab [8, 9, 10] has played a central role in advancing research on automated fact-checking by organizing shared tasks that support the development of tools and datasets for these sub-tasks across languages, domains, and media types.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
Dhruv.Sahnan@mbzuai.ac.ae (D. Sahnan); tanchak@iitd.ac.in (T. Chakraborty); Preslav.Nakov@mbzuai.ac.ae (P. Nakov)
0000-0002-5205-8269 (D. Sahnan); 0000-0002-0210-0369 (T. Chakraborty); 0000-0002-3600-1510 (P. Nakov)

[1]Newtral.es: https://www.newtral.es; Full Fact: https://fullfact.org; Pagella Politica: https://pagellapolitica.it

Recently, however, Sahnan et al. [11] argued that the automatic fact-checking pipeline must be extended with an additional task: the automatic generation of full fact-checking articles. This task is especially important in the professional fact-checking workflow, where fact-checkers publish the final article, which is not merely a label or a short justification, but the main artifact through which readers understand why a claim is true, false, misleading, or otherwise nuanced. Yet, while previous editions of the CheckThat! lab have addressed the five core sub-tasks listed above, the important and time-consuming task of writing full fact-checking articles that guide readers toward a comprehensive understanding of a claim has remained largely unaddressed. Therefore, in the 2026 edition of the CLEF CheckThat! lab, we introduce, for the first time, a shared task on automatically generating full fact-checking articles, with the goal of advancing research on both generation methods and their evaluation. In this paper, we present a detailed overview of the task, which is defined as follows: *given a claim, its veracity label, and a set of evidence documents consulted during the fact-checking process, the goal is to generate a full fact-checking article with inline citations.*

We evaluate participating systems using a dual-focus evaluation strategy. First, the official leaderboard is based on automatic evaluation measures that assess generated articles in terms of entailment, citation recall, citation precision, and evidence coverage. These metrics are designed to measure whether generated articles cover all of the information in the reference articles, whether they are properly grounded in the evidence, and whether each factual statement is cited with the relevant evidence documents. Second, we conduct an additional LLM-as-a-judge evaluation to assess dimensions that are harder to capture with reference-based or citation-focused metrics alone, such as writing quality, coherence, clarity, and comprehensibility. We use LLM-as-a-judge to compare the outputs of participating systems in a pairwise setup and rank them using Elo ratings. Together, these two evaluation perspectives allow us to analyze systems both in terms of evidence grounding and in terms of their ability to produce clear, coherent, and easy to understand articles that follow professional standards of writing.

In the following sections, we describe the datasets used in the task, the evaluation methodology, the participating systems and their approaches, and our discussion on the insights from this first shared task on full fact-checking article generation.

2. Background

Automated fact-checking has traditionally decomposed professional fact-checking into a set of sub-tasks designed to assist human experts [4, 12, 5, 7]. These include detecting check-worthy claims [13, 14, 15, 16], identifying previously fact-checked claims [17], retrieving relevant evidence [18, 19, 10], predicting claim veracity [20, 21, 22], and generating explanations for verification decisions [23, 12, 24, 25]. While these components capture important parts of the verification process, they do not fully reflect the output produced by professional fact-checkers. In practice, fact-checkers do not only assign a verdict to a claim; they also write a public-facing fact-checking article that contextualizes the claim being checked, clarifies ambiguities, traces the claim's origin, explains the evidence, and guides the readers toward an informed understanding of both the claim and the broader narrative surrounding it [26, 11, 27, 28]. Despite its importance, this final article-writing step remains largely unaddressed in automatic fact-checking pipelines, even though it is a central and time-consuming part of the professional workflow.

Recent advances in large language models (LLMs) have opened possibilities for automating this stage of the fact-checking workflow, i.e., automatically generating the full fact-checking article. As such, Sahnan et al. [11] recently proposed using LLMs to generate full fact-checking articles for a given claim, moving beyond short explanations or verdict-oriented outputs. This line of work frames fact-checking article generation as a grounded long-form generation task, where the system must synthesize the evidence consulted for the fact-check into an article suitable to be published for the public. Furthermore, prior human-centered analyses on professional fact-checking highlight several requirements for such articles, including factual accuracy, transparent use of sources, clear reasoning over multiple pieces of evidence, and explanations that help readers understand how the evidence supports or contradicts the claim [11, 27, 28].

Transparency is particularly important in this setting, as readers should be able to inspect the sources used in the verification process and assess whether the evidence is correctly represented. For this reason, systems for fact-checking article generation in the CLEF 2026 CheckThat! lab are expected not only to produce a coherent fact-checking article, but also to include inline citations that connect factual statements in the generated article to the relevant evidence.

In previous work, Sahnan et al. [11] proposed *QRAFT* for this task of generating full fact-checking articles, which is a multi-agent pipeline that mimics the article writing workflow of professional fact-checkers. Rather than generating the article in a single step, QRAFT decomposes the process into stages that support evidence understanding, article planning, drafting, and revision based on an editorial review. To the best of our knowledge, QRAFT remains the only specialized pipeline proposed specifically for full fact-checking article generation. More broadly, the task is related to long-form generation and expository text generation. Long-form generation is challenging because the output must remain coherent across multiple paragraphs, stay focused on the topic, and follow a clear structure. These challenges are especially pronounced in expository writing, where the goal is to explain a topic clearly and accurately on the basis of external information, and preserve factual consistency throughout the text. Balepur et al. [29] proposed *IRP*, an approach that decomposes generation into iterative steps of planning, retrieval, and paraphrasing, reflecting the need to first identify relevant facts and then organize them into a coherent explanation. Other work has explored similar ideas for different long-form genres. Shao et al. [30] studied the generation of Wikipedia-like articles from scratch, and proposed *Storm*, which emphasizes the importance of a pre-writing stage where it researches the topic, collects information from multiple perspectives, and constructs an outline before drafting the article. Recent work on specialized long-form generation has similarly emphasized a pre-writing / planning stage, for e.g., Wang et al. [31] proposed *AutoPatent*, a multi-agent framework for generating complete patent documents, and Wang et al. [32] proposed *AutoSurvey* to address the generation of literature surveys from large collection of research papers. In addition, Liu and Chang [33] introduced exemplar-based expository text generation and proposed *RePA*, a system which generates an explanatory text for a new topic by adapting the structure and style of an exemplar document while preserving coherence and factual relevance.

These studies provide useful priors for building fact-checking article generation pipelines. However, fact-checking articles impose additional constraints that are not fully covered by general long-form or expository generation tasks. A fact-checking article generation system must produce a well-structured article reasoning over potentially conflicting evidence, explaining the relation between the evidence and the claim, communicating the verdict, and providing citations that make the generated article transparent and verifiable. Overall, despite its importance in the professional fact-checking workflow, this task of automatically generating full fact-checking articles is relatively new and there has been little exploration into how different generation pipelines perform under identical settings that involve common data, evidence sets, and metrics. To address this gap, we organized Task 3 in the CLEF 2026 CheckThat! lab, on generating full fact-checking articles, with the goal of evaluating how well systems can produce complete, evidence-grounded fact-checking articles from a given claim, its verdict and a set of evidence documents consulted for the fact-check.

3. Data

We build the data for the task from three fact-checking corpora suitable for long-form article generation, namely, *WatClaimCheck* [34], *ExClaim* [25], and *AmbiguousSnopes*. WatClaimCheck and ExClaim were introduced in prior work, while AmbiguousSnopes is newly released as part of this task. Below, we provide a brief overview of the training, development, and test splits, constructed using these three corpora.

Training and development splits. We reuse the training and development splits from the WatClaimCheck dataset to form the corresponding splits for this task.

Table 1
Statistics about the training, development, and test splits for the Task 3 dataset. For each split, we present the comprising datasets, the number of examples, and the dates between which the fact-checks were originally published on their respective websites.

Split	Comprising Datasets	# Examples	Published Between
Training	WatClaimCheck training split	26976	13 Dec 1996 — 28 Mar 2021
Development	WatClaimCheck development split	3372	02 Nov 1997 — 27 Mar 2021
Test	ExClaim test split	987	30 Oct 2008 — 27 Mar 2021
	AmbiguousSnopes	171	16 Feb 2022 — 30 Aug 2024

WatClaimCheck comprises fact-checked claims collected from several public fact-checking websites, together with the original verdicts assigned by professional fact-checkers, the full expert-written fact-checking articles, and the evidence documents used to verify each claim and draft the corresponding article. In total, the training and development splits contain approximately 26k and 3.3k examples, respectively.

Test split. We use a union of the ExClaim and AmbiguousSnopes datasets to construct the test split. ExClaim is a smaller subset of the WatClaimCheck test split, introduced by Zeng and Gao [25], which we use rather than the full WatClaimCheck test split because generating and evaluating long-form articles is computationally expensive. On the other hand, AmbiguousSnopes is a relatively fresher collection of hard claims drawn from Snopes[2], for which the available evidence does not support a simple binary True / False judgement. This focused dataset was curated following feedback from fact-checking experts, who emphasized that real-world fact-checking systems must also handle claims that require a nuanced, in-depth analysis. Overall, merging the two datasets yields 1158 examples in the test split: 987 from ExClaim and 171 from AmbiguousSnopes. Each example in this split follows the same format as those in the training and development splits, comprising the claim, a veracity label, the evidence documents used to verify the claim and draft the article, and the corresponding expert-written fact-checking article. We treat the expert-written article as the reference output for the task.

We further present some statistics about each data split in Table 1, which includes the specific datasets used, the number of examples in each dataset, and the dates between which the fact-checking articles were published on the respective websites.

4. Evaluation Setup

Our task is one of evidence-grounded long-form generation, which expects full fact-checking articles to be generated with inline citations to the evidence. Previous work has argued that evaluating free-form generated text is hard, particularly when the outputs are long form and must be grounded in evidence [35, 36, 37]. Xu et al. [38] also proposed a multi-faceted evaluation setup for long-form generated text, since a single score often does not indicate text quality on fine-grained aspects such as factuality, completeness, coherence, and practical usefulness. Such challenges in evaluation are particularly pronounced for fact-checking article generation, where practical usefulness depends not only on fluency or similarity to a reference article, but also on factual accuracy, proper and transparent use of evidence, quality of the reasoning to explain the claim, and adherence to professional fact-checking writing standards [11]. Therefore, for ranking participants in terms of their performance, we designed our automatic evaluation to focus on the following aspects: *information consistency*, *information coverage*, and *evidence grounding*. Additionally, we conducted secondary evaluations using LLM-as-a-judge to assess the *writing quality* of the articles generated by the participants.

[2] https://www.snopes.com/fact-check/

Automatic evaluation measures. We report four measures: *(i)* Entailment Score (ES), a bidirectional chunk-level entailment measure between the generated and reference articles that estimates whether the generated article is both consistent with and covers the reference content [11]; *(ii)* Citation Recall (CR) and *(iii)* Citation Precision (CP), following prior work on cited and verifiable generation [39], which assess whether generated statements are supported by their inline citations and whether the cited evidence is relevant; and (iv) Evidence Coverage (EC), which computes the proportion of input evidence documents that are cited in the generated article. We use a simple average of the four measures to rank teams on the official task leaderboard.

LLM-as-a-judge evaluation strategy. Evaluating the writing quality of the generated full fact-checking articles is challenging because the output text is long-form and open-ended, and therefore cannot be fully captured by the automatic evaluation measures described above. Recent work has shown that LLM-based judges can provide scalable assessments of open-ended generations that correlate with human preferences across diverse generation tasks [40, 36, 41, 42], while pairwise comparison frameworks offer a practical way to aggregate relative preferences into system-level rankings using Elo ratings [40, 43, 44]. Motivated by this line of work, we adopt an LLM-as-a-judge evaluation strategy to compare participating systems in terms of writing style, and rank them using Elo ratings. Specifically, we use `Prometheus-2 (7B)` [45] as the rater in a relative grading setup, where the model is presented with articles generated by two participating systems and asked to select the better one. The articles are compared based on whether they are well written, clear, and easy to understand. We further prompt the model to focus on several aspects: whether the claim is clearly stated and sufficiently contextualized; whether the article is well-structured and coherent; whether it is concise while preserving necessary detail; whether the language is clear and accessible to a broad audience; and whether the writing style is appropriate for fact-checking, polished, and professional. Finally, based on these pairwise grading decisions, we compute Elo ratings for each participant and use them to rank teams on a secondary evaluation leaderboard. We set the starting Elo rating for each system at 1400.

5. Official Task Leaderboard and the Participating Systems

We received submissions from a total of 10 teams, and one additional team that submitted post the task competition deadline. Out of these, 8 teams submitted their working notes. We provide an overview of the various participating systems along with the official task leaderboard in Table 2. Team AKSH, who submitted their generations after the competition ended is left unranked. Furthermore, team Facty, team Ro and team Shtian did not submit their working notes, and their approaches are thus left undiscussed.

Baseline. We use `GPT-5 Nano` [54] as our baseline. Given the claim and the associated evidence documents, we first segment each document into chunks of five sentences each. Then for each document, we fetch the top-k (we set $k = 5$) most relevant chunks through dense retrieval, where we use OpenAI's `text-embedding-3-large` as the embedding model and use cosine similarity between the text embedding of the claim and the text embedding of the evidence chunks to compute relevance. Finally, we prompt `GPT-5 Nano` to generate a full fact-checking article with inline citations, given the claim, its veracity label, and the retrieved evidence chunks. Notably, 10 out of the 11 teams that submitted their generations beat the baseline, with only team Shtian being the exception.

5.1. Overview of the Participating Systems

Most teams chose to navigate this task in two main stages: first, they processed the input evidence to focus on information most relevant to the claim; and second, they used the processed evidence to generate the full fact-checking article. For the input evidence processing stage, teams used several techniques including but not limited to dense retrieval, re-ranking evidence based on relevance, and LLM-based retrieval of relevant facts from evidence.

Table 2
Official task leaderboard and overview of the participating systems. The teams are ranked based on average (Avg.) score across four measures: Entailment Score (ES), Citation Recall (CR), Citation Precision (CP), and Evidence Coverage (EC). "↑" indicates higher scores are better. **Bold scores** represent best performance on that evaluation measure.

Rank	Team	Approach												Performance				
		Evidence Processing							Generation									
		LLM Prompting	Summarization	Dense Retrieval	Re-ranking	Cosine Similarity	Entailment	Other	LLM Prompting	Multi-Step	Multi-Agent	Few-Shot	CoT Enabled	Other	Avg.	ES ↑	CR ↑	CP ↑	EC ↑
1	Outsider [46]	■							■						**0.55**	0.28	**0.67**	**0.67**	0.56
2	UTS [47]						■	■						■	0.48	0.34	0.30	0.30	**0.99**
3	DS@GT [48]			■		■		■	■						0.43	0.29	0.28	0.24	0.90
4	Facty														0.43	0.23	0.25	0.23	0.99
5	UCPH RMT [49]	■	■						■	■					0.42	**0.36**	0.17	0.14	0.99
6	Ro														0.40	0.32	0.46	0.46	0.34
7	SourceMinds [50]			■	■	■			■		■				0.33	0.24	0.34	0.34	0.39
8	CheckMate [51]	■							■						0.33	0.22	0.35	0.35	0.39
9	KNU Fact [52]			■			■		■				■		0.31	0.23	0.29	0.29	0.44
–	Baseline			■		■			■						0.27	0.30	0.24	0.22	0.33
10	Shtian														0.22	0.28	0.20	0.19	0.21
–	AKSH [53]	■							■			■			0.38	0.33	0.32	0.30	0.56

The generation setup also varied across teams ranging from simple LLM-based prompting or multi-step and multi-agent generation pipelines, to heuristic-driven concatenation of relevant snippets from the input evidence. Some teams further incorporated post-processing steps to ensure that the final fact-checking article included citations for each factual statement and that these citations conformed to the format required by the task's evaluation script.

Team **Outsider** [46], the top-ranked team, achieved the best average score of 0.546, driven by high performance on citation-oriented metrics with a score of 0.671 on both citation recall and citation precision. In their approach, they first used `Llama-3.2-1B` to extract concise factual statements from the provided evidence documents, and then prompted `Mistral-Small-4-119B` to generate the article using these statements. Finally, they also appended the extracted factual statements in the format "`<sentence> (source: <url>)`" as a list of the evidence used during generation to align the output for the citation-based evaluation.

Team **UTS** [47] ranked second, achieving near-perfect evidence coverage (0.996). Interestingly, rather than using a language model for free-form generation to draft the article, they used a deterministic citation-text engineering strategy. They converted each sentence in the given evidence documents into a templated citation-text format "`<sentence> (source: <url>)`", and then used `Llama-3.2-1B` to verify if each citation-text passed the citation entailment check required for measuring citation recall and citation precision by the task's scoring script. They simply concatenated these citation-texts together to form the full fact-checking article.

Team **DS@GT** [48] ranked third with an approach that focused on preprocessing and organizing evidence before generation. They first removed empty and noisy evidence documents with no relevant information about the claim from the input evidence set, before cleaning the usable documents in a preprocessing step to remove HTML/JavaScript tags and boilerplate text. Then, they filtered evidence based on cosine similarity against the claim using sentence embeddings, and clustered together evidence that discusses the same fact or closely related facts. Finally, they used `Llama-3.1-8B-Instruct` to generate the full fact-checking article given the claim, the original verdict, and the clustered evidence documents.

Team **UCPH RMT** [49] achieved the highest entailment score on the test dataset. They chose a multi-step generation paradigm where they first chunked each evidence documents and generated claim-aware summaries for each chunk. In turn for each evidence document, they used `Llama-3.1-8B-Instruct` to produce a paragraph analysing the claim using claim-aware summaries for only that evidence document. Finally, they appended the citation string: "`(source: <url>)`" to each generated paragraph and concatenated all such paragraphs to form the full fact-checking article.

Team **SourceMinds** [50] proposed a pipeline with a dense retrieval step followed by multi-agent generation using LLMs. They first retrieved relevant passages with `sentence-transformers/all-mpnet-base-v2` and reranked claim-evidence pairs using the `BAAI/bge-reranker-v2-m3` cross-encoder. In their generation workflow, they used `Qwen2.5-32B-Instruct` and it comprised three agents: a planner to prepare a plan using the retrieved evidence, a writer to drafting an article using the plan, and a self-critique agent to revise the article draft based on certain heuristics. They also implemented a post processing citation audit step to ensure the all factual assertions were supported by some input evidence.

Team **CheckMate** [51] also focused heavily on processing the given evidence documents to extract relevant information before generating the final article. They used `gemini-2.5-flash` to first identify usable evidence documents and rejected those with noisy or irrelevant text. Then, they used `gemini-3-flash-preview` to extract information relevant to the claim along with some additional metadata relevant for the analysis. This evidence was also augmented with additional flags to distinguish important evidence from background or irrelevant material. Finally, they used `gemini-3-flash-preview` to generate the full fact-checking article using this processed evidence. To control citations, the generator model referred to sources through temporary evidence identifiers, which were later replaced with the exact URLs and validated using Python post-processing.

Team **KNU Fact** [52] also proposed a pipeline that centered on filtering the evidence to the most relevant chunks. Their pipeline first chunked each evidence document and used entailment-based filtering to select the top relevant chunks, including both entailing and contradicting evidence. They then used the filtered evidence to prompt an LLM to draft the full fact-checking article. Notably, they used Chain-of-Thought prompting on a thinking-enabled LLM for generation with the goal of improved evidence aggregation, evidence grounding and overall article quality.

Team **AKSH** [53], listed as an unranked submission, proposed a pipeline that used a few-shot article generation strategy. Their pipeline first extracted useful facts from the given evidence and removed boilerplate text, and then prompted an LLM to generate structured articles using a sample fact-checking article as an in context example. They also implemented a post-processing step to add missing citations to factual sentences.

5.2. LLM-as-a-Judge Evaluation

We further perform LLM-as-a-judge evaluations for a secondary view of system performance that complements the automatic evaluation used for the official task leaderboard. While the official leaderboard in Table 2 ranks systems based on entailment, citation recall, citation precision, and evidence coverage, these measures do not fully capture the quality of the generated article as a public-facing output of the fact-checking workflow. In particular, a system may cite many evidence documents or satisfy citation-oriented metrics, while still producing an article that is difficult to read. Conversely, a system may produce a fluent and coherent article, but one that does not cite any evidence for factual statements or that contains hallucinated content. The secondary leaderboard in Table 3 should therefore be interpreted as an additional evaluation perspective, focused on writing quality, coherence, clarity, and readability.

Overall, Table 3 shows that the Elo-based ranking differs substantially from the official task leaderboard, with only team DS@GT retaining its rank in both leaderboards. The strongest systems according to the LLM-as-a-judge evaluation are CheckMate, KNU Fact, and DS@GT: these systems share a broad design pattern, in which they first processed the provided evidence to identify useful or relevant information, and then used an LLM to generate a full article from the processed evidence.

Table 3
Secondary leaderboard based on LLM-as-a-judge Elo ratings for writing quality. Teams are ranked by Elo ratings, where higher scores indicate stronger performance (↑). The **bold score** represents the best performing team. We also show ΔRank, which represents the change in ranking in this leaderboard as compared to the official task leaderboard.

Rank	Team	Elo Rating ↑	ΔRank
1	CheckMate [51]	**1647.35**	↑ 7
2	KNU Fact [52]	1631.27	↑ 7
3	DS@GT [48]	1592.32	–
4	Outsider [46]	1543.48	↓ 3
5	Facty	1503.89	↓ 1
6	SourceMinds [50]	1358.52	↑ 1
7	Shtian	1311.89	↑ 3
8	Ro	1229.83	↓ 2
9	UCPH RMT [49]	1107.72	↓ 4
10	UTS [47]	999.65	↓ 8
–	AKSH [53]	1474.10	–

This suggests that systems that combine evidence processing with fluent LLM-based article drafting emerge competitive on writing quality. Notably, team CheckMate and team KNU Fact show the largest improvements, each moving up by seven positions. Team CheckMate filtered and extracted relevant information from evidence documents before generating the final article, and then validated and replaced temporary source identifiers during post-processing. Team KNU Fact used entailment-based evidence filtering and Chain-of-Thought prompting to support evidence aggregation and article generation. Their strong performance in the Elo-based evaluation again reinforces that coherent and polished articles produced by LLMs are rewarded more by the judges. Finally, team Outsider, the official winner, drops to fourth place but remains competitive, suggesting that its strong citation-oriented performance is accompanied by reasonably strong article quality.

On the other hand, teams Ro, UCPH RMT, and UTS achieve the lowest Elo ratings among the participating systems. Notably, team UTS observes the most striking change in its rankings across the two leaderboards, moving from second place in the official task leaderboard to last place in the LLM-as-a-judge evaluation setup. This large drop can be explained by the nature of their approach. Team UTS used a deterministic citation-text engineering strategy: sentences from the evidence documents were converted into citation-text units and concatenated to form the final output. This approach was highly effective for the citation-oriented metrics as well as evidence coverage, where team UTS achieved a near-perfect score. However, concatenating citation-text units does not naturally produce a coherent long-form article with a clear narrative, smooth transitions, or an explanatory structure. Team UCPH RMT also observes a drop from fifth place in the official task leaderboard to ninth place in the LLM-as-a-judge rankings due to a similar reason. In their approach, they generated claim-aware analysis paragraphs for each document that were then concatenated into the final article. Again, this concatenation of independent paragraphs may produce an output that reads more like a sequence of evidence-based paragraphs than a coherent fact-checking article with a clear narrative. This contrast illustrates that content overlap or entailment with the reference does not always translate into strong article-level readability. Thus, the same design choices that made the systems for teams UTS and UCPH RMT strong under automatic citation and coverage metrics likely made them less competitive in an evaluation focused on writing quality.

6. Discussion

The participating systems tried several strategies for generating full fact-checking articles, which typically included two main stages: processing the input evidence; and using the processed evidence to generate the fact-checking article. Some teams also included a post-processing step to ensure all factual statements are cited with some evidence. Therefore, we analyze the participants' approaches through three dimensions below: evidence processing, the generation step, and the post-processing strategy.

6.1. Evidence Processing

Processing the input evidence to focus on the key facts and statements relevant to a given claim emerged as a common step across participating systems. This was largely motivated by the nature of the evidence documents as they were often long, contained noisy elements such as HTML tags and boilerplate text originating from the source webpages, and sometimes included information that was only loosely related or entirely irrelevant to the input claim. As a result, evidence processing was an important intermediate step, which helped teams construct a more compact and focused representation of the available evidence before drafting the fact-checking article.

Teams adopted a range of strategies for this purpose. Some teams used retrieval-based approaches, including dense retrieval and passage re-ranking to identify the evidence passages most relevant to the claim or to its constituent sub-claims. Others used LLM-based claim-aware fact extraction or summarization to identify the information most relevant to the claim and to support the final verdict and explanation. Overall, this design choice in these approaches suggests that the quality of the generated article depends not only on the generative model itself, but also on how effectively the available evidence is cleaned, filtered, and structured before generation.

6.2. Generation

For generating the full fact-checking articles, several teams prompted general-purpose, off-the-shelf LLMs using the processed evidence. The models used by participating systems included `Mistral-Small-4-119B`, `Llama-3.1-8B-Instruct`, `Qwen2.5-32B-Instruct`, `Gemini-3-Flash-Preview`, and `Qwen3-8B`, among others. These systems typically framed article generation as an evidence-grounded long-form generation task, instructing the model to synthesize the filtered evidence into a coherent verdict, explanation, and set of supporting citations. Teams implemented this LLM-based generation stage in different ways: some used simple prompting setups to generate the article directly, others adopted multi-agent pipelines in which different agents handled planning, outlining, and drafting, and some used chain-of-thought prompting or thinking modes in reasoning models to elicit intermediate reasoning before producing the final article.

Other teams adopted more modular strategies, creating citation texts from sentences in the evidence documents, or generating independent evidence-based analyses using LLMs before stitching them together to form an article. These approaches performed well on the automatic metrics used in the official leaderboard, particularly citation-oriented metrics, since they often preserved strong local links between generated text and supporting evidence. However, they were rated less favorably by the LLM-as-a-judge as independently generated evidence-based passages when stitched together, do not necessarily form a coherent and easily understandable long-form article. In contrast, systems relying more directly on LLM-based article generation tended to receive higher LLM-as-a-judge ratings, resulting in higher rankings in Table 3. This suggests that while modular generation can be effective for optimizing for citation-oriented measures, LLM-based generation may be more coherent and readable, which is expected of a publishable full fact-checking article for a general audience. This indicates a need to develop a comprehensive evaluation framework that jointly evaluates evidence grounding, proper citations, and information coverage as well as coherence, readability, and usefulness of the generated articles, in line with experts' needs.

6.3. Post-Processing

Finally, several teams also incorporated post-processing steps after article generation to further improve performance on the citation-oriented measures and ensure compatibility with the task evaluation setup. These steps included adding citations to factual statements that were missing explicit source links, replacing temporary evidence identifiers with the corresponding URLs as provided in the input, validating citation formats, and appending evidence-derived factual statements at the end of the generated article.

These post-processing strategies were particularly useful for improving citation-oriented automatic metrics. For example, appending evidence-backed factual statements or templated citation texts increased the likelihood that generated content was associated with a verifiable source, thereby improving citation recall and citation precision. Similarly, including a broader list of cited evidence could improve evidence coverage (as evident by near perfect performance by several teams in Table 2) by ensuring that all or most of the input evidence documents were represented in the final output. However, these gains did not always correspond to better article quality. Outputs that included appended evidence lists or mechanically inserted citations could score well under automatic citation and coverage metrics, yet, they may not be what professional fact-checkers expect in a publishable article. This again highlights the need to develop evaluation frameworks that move beyond rewarding simple optimizations and focus more on rewarding articles that actually align with professional standards of writing.

7. Conclusion

We presented a detailed overview of Task 3 of the CheckThat! Lab at CLEF 2026, which addressed the generation of full fact-checking articles with inline citations. The task attracted submissions from 10 teams, along with one additional team submitting after the official competition deadline. The participating systems typically approached the task using a two-stage pipeline, in which they first processed the input evidence to identify the most relevant information through dense retrieval, reranking, and LLM-based fact extraction or summarization; and then used this focused evidence set to generate the final fact-checking article through vanilla LLM prompting, multi-step and multi-agent setups for planned long-form generation, or even simply concatenating relevant evidence snippets. Our dual-focus evaluation strategy provided useful insight into these systems from two complementary perspectives, that is, evidence grounding and proper citation quality on the one hand, and the writing quality expected of professional fact-checking articles on the other. This task extends the traditional CheckThat! lab pipeline by introducing an additional step of automatically generating public-facing fact-checking articles aligned with the needs of professional fact-checkers.

In future work, we aim to focus on improving generation, better aligning automatic and LLM-based evaluation with expert needs, and encouraging systems that balance evidence grounding, information coverage, coherence, readability, and transparency. We also plan to explore task setups that allow for expert intervention, enabling human–AI collaboration to guide models toward more reliable and useful fact-checking article generation.

Acknowledgments

The authors would like to acknowledge the Multi-Institutional Faculty Interdisciplinary Research Project (MFIRP) between IIT Delhi and MBZUAI and Anusandhan National Research Foundation (DST/INT/USA/NSF-DST/Tanmoy/P-2/2024) for financial support. T.C. further acknowledges the support of the Rajiv Khemani Young Faculty Chair Professorship in Artificial Intelligence.

Declaration on Generative AI

In this study, we employed `GPT-5 Nano` and prompted it using generic instructions for generating full fact-checking articles with inline citations to serve as the baseline system. Moreover, the author(s) used `GPT-5` for grammar and spell check through an enterprise version of the ChatGPT application. After using this tool, the author(s) reviewed and edited the AI-suggested content as needed. Generative AI was not used to generate the content of the main manuscript. The authors take full responsibility for the final content of the publication.

References

[1] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, Journal of Economic Perspectives 31 (2017) 211–36. URL: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211. doi:10.1257/jep.31.2.211.

[2] M. Minici, F. Cinus, L. Luceri, E. Ferrara, Uncovering coordinated cross-platform information operations: Threatening the integrity of the 2024 U.S. presidential election, First Monday 29 (2024). URL: https://firstmonday.org/ojs/index.php/fm/article/view/13831. doi:10.5210/fm.v29i11.13831.

[3] F. Alam, S. Shaar, F. Dalvi, H. Sajjad, A. Nikolov, H. Mubarak, G. Da San Martino, A. Abdelali, N. Durrani, K. Darwish, A. Al-Homaid, W. Zaghouani, T. Caselli, G. Danoe, F. Stolk, B. Bruntink, P. Nakov, Fighting the COVID-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 611–649. URL: https://aclanthology.org/2021.findings-emnlp.56/. doi:10.18653/v1/2021.findings-emnlp.56.

[4] 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. URL: https://aclanthology.org/W14-2508. doi:10.3115/v1/W14-2508.

[5] P. Nakov, D. Corney, M. Hasanain, F. Alam, T. Elsayed, A. Barrón-Cedeño, P. Papotti, S. Shaar, G. Da San Martino, Automated fact-checking for assisting human fact-checkers, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 4551–4558. URL: https://doi.org/10.24963/ijcai.2021/619. doi:10.24963/ijcai.2021/619, survey Track.

[6] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news, in: Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2021, p. 639–649. URL: https://doi.org/10.1007/978-3-030-72240-1_75. doi:10.1007/978-3-030-72240-1_75.

[7] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206. URL: https://aclanthology.org/2022.tacl-1.11. doi:10.1162/tacl_a_00454.

[8] A. Barrón-Cedeño, F. Alam, J. M. Struß, P. Nakov, T. Chakraborty, T. Elsayed, P. Przybyła, T. Caselli, G. Da San Martino, F. Haouari, C. Li, J. Piskorski, F. Ruggeri, X. Song, R. Suwaileh, Overview of the CLEF-2024 CheckThat! Lab: Check-worthiness, subjectivity, persuasion, roles, authorities and adversarial robustness, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[9] F. Alam, J. M. Struß, T. Chakraborty, S. Dietze, S. Hafid, K. Korre, A. Muti, P. Nakov, F. Ruggeri, S. Schellhammer, V. Setty, M. Sundriyal, K. Todorov, V. Venktesh, Overview of the CLEF-2025 CheckThat! Lab: Subjectivity, fact-checking, claim normalization, and retrieval, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[10] J. M. Struß, S. Schellhammer, S. Dietze, V. V., V. Setty, T. Chakraborty, P. Nakov, A. Anand, P. Chungkham, S. Hafid, D. Sahnan, K. Todorov, Overview of the CLEF-2026 CheckThat! Lab: Advancing multilingual fact-checking, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer, 2026.

[11] D. Sahnan, D. Corney, I. Larraz, G. Zagni, R. Miguez, Z. Xie, I. Gurevych, E. Churchill, T. Chakraborty, P. Nakov, Can LLMs automate fact-checking article writing?, Transactions of the Association for Computational Linguistics 14 (2026) 489–509. URL: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/TACL.a.644/2597081/tacl.a.644.pdf. doi:10.1162/TACL.a.644.

[12] N. Kotonya, F. Toni, Explainable automated fact-checking: A survey, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 5430–5443. URL: https://aclanthology.org/2020.coling-main.474. doi:10.18653/v1/2020.coling-main.474.

[13] N. Hassan, F. Arslan, C. Li, M. Tremayne, Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 1803–1812. URL: https://doi.org/10.1145/3097983.3098131. doi:10.1145/3097983.3098131.

[14] D. Wright, I. Augenstein, Claim check-worthiness detection as positive unlabelled learning, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 476–488. URL: https://aclanthology.org/2020.findings-emnlp.43/. doi:10.18653/v1/2020.findings-emnlp.43.

[15] L. Konstantinovskiy, O. Price, M. Babakar, A. Zubiaga, Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection, Digital Threats 2 (2021). URL: https://doi.org/10.1145/3412869. doi:10.1145/3412869.

[16] A. Barrón-Cedeño, F. Alam, A. Galassi, G. Da San Martino, P. Nakov, , T. Elsayed, D. Azizov, T. Caselli, G. Cheema, F. Haouari, M. Hasanain, M. Kutlu, C. Li, F. Ruggeri, J. M. Struß, W. Zaghouani, Overview of the CLEF–2023 CheckThat! Lab checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), 2023.

[17] S. Shaar, N. Babulkov, G. Da San Martino, P. Nakov, That is a known lie: Detecting previously fact-checked claims, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL '20, Association for Computational Linguistics, Online, 2020, pp. 3607–3618. URL: https://aclanthology.org/2020.acl-main.332. doi:10.18653/v1/2020.acl-main.332.

[18] A. Soleimani, C. Monz, M. Worring, BERT for evidence retrieval and claim verification, in: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2020, p. 359–366. URL: https://doi.org/10.1007/978-3-030-45442-5_45. doi:10.1007/978-3-030-45442-5_45.

[19] A. Zou, Z. Zhang, H. Zhao, Decker: Double check with heterogeneous knowledge for commonsense fact verification, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 11891–11904. URL: https://aclanthology.org/2023.findings-acl.752. doi:10.18653/v1/2023.findings-acl.752.

[20] I. Augenstein, C. Lioma, D. Wang, L. Chaves Lima, C. Hansen, C. Hansen, J. G. Simonsen, MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 4685–4697. URL: https://aclanthology.org/D19-1475. doi:10.18653/v1/D19-1475.

[21] M. S. Schlichtkrull, Z. Guo, A. Vlachos, AVeriTeC: A dataset for real-world claim verification with evidence from the web, in: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, NeurIPS '23, 2023. URL: https://openreview.net/forum?id=fKzSz0oyaI.

[22] Z. Xie, R. Xing, Y. Wang, J. Geng, H. Iqbal, D. Sahnan, I. Gurevych, P. Nakov, FIRE: Fact-checking with iterative retrieval and verification, in: Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 2901–2914. URL: https://aclanthology.org/2025.findings-naacl.158/. doi:10.18653/v1/2025.findings-naacl.158.

[23] 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, ACL '20, Association for Computational Linguistics, Online, 2020, pp. 7352–7364. URL: https:

//aclanthology.org/2020.acl-main.656. doi:10.18653/v1/2020.acl-main.656.
[24] D. Russo, S. S. Tekiroğlu, M. Guerini, Benchmarking the generation of fact checking explanations, Transactions of the Association for Computational Linguistics 11 (2023) 1250–1264. URL: https://aclanthology.org/2023.tacl-1.71/. doi:10.1162/tacl_a_00601.
[25] 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. URL: https://aclanthology.org/2024.tacl-1.19. doi:10.1162/tacl_a_00649.
[26] L. Graves, Anatomy of a fact check: Objective practice and the contested epistemology of fact checking, Communication, Culture and Critique 10 (2017) 518–537. URL: https://doi.org/10.1111/cccr.12163. doi:10.1111/cccr.12163.
[27] H. Liu, A. Das, A. Boltz, D. Zhou, D. Pinaroc, M. Lease, M. K. Lee, Human-centered NLP fact-checking: Co-designing with fact-checkers using Matchmaking for AI, Proceedings of the ACM on Human-Computer Interaction 8 (2024). URL: https://doi.org/10.1145/3686962. doi:10.1145/3686962.
[28] G. Warren, I. Shklovski, I. Augenstein, Show me the work: Fact-checkers' requirements for explainable automated fact-checking, in: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA, 2025. URL: https://doi.org/10.1145/3706598.3713277.
[29] N. Balepur, J. Huang, K. Chang, Expository text generation: Imitate, retrieve, paraphrase, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP '23, Association for Computational Linguistics, Singapore, 2023, pp. 11896–11919. URL: https://aclanthology.org/2023.emnlp-main.729. doi:10.18653/v1/2023.emnlp-main.729.
[30] Y. Shao, Y. Jiang, T. Kanell, P. Xu, O. Khattab, M. Lam, Assisting in writing Wikipedia-like articles from scratch with large language models, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), ACL '24, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 6252–6278. URL: https://aclanthology.org/2024.naacl-long.347.
[31] Q. Wang, S. Ni, H. Liu, S. Lu, G. Chen, X. Feng, C. Wei, Q. Qu, H. Alinejad-Rokny, Y. Lin, et al., AutoPatent: A multi-agent framework for automatic patent generation, ArXiv preprint abs/2412.09796 (2024). URL: https://arxiv.org/abs/2412.09796.
[32] Y. Wang, Q. Guo, W. Yao, H. Zhang, X. Zhang, Z. Wu, M. Zhang, X. Dai, M. zhang, Q. Wen, W. Ye, S. Zhang, Y. Zhang, AutoSurvey: Large language models can automatically write surveys, in: Advances in Neural Information Processing Systems, volume 37 of *NeurIPS '24*, Curran Associates, Inc., 2024, p. 115119–115145. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/d07a9fc7da2e2ec0574c38d5f504d105-Paper-Conference.pdf.
[33] Y. Liu, K. C.-C. Chang, Writing like the best: Exemplar-based expository text generation, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL '25, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 25739–25764. URL: https://aclanthology.org/2025.acl-long.1250/. doi:10.18653/v1/2025.acl-long.1250.
[34] 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), ACL '22, Association for Computational Linguistics, Dublin, Ireland, 2022, p. 1293–1304. URL: https://aclanthology.org/2022.acl-long.92. doi:10.18653/v1/2022.acl-long.92.
[35] K. Krishna, A. Roy, M. Iyyer, Hurdles to progress in long-form question answering, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL '21, Association for Computational Linguistics, Online, 2021, pp. 4940–4957. URL: https://aclanthology.org/2021.naacl-main.393/. doi:10.18653/v1/2021.naacl-main.393.
[36] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-Eval: NLG evaluation using GPT-4 with better human alignment, in: Proceedings of the 2023 Conference on Empirical Methods in Natural

Language Processing, EMNLP '23, Association for Computational Linguistics, Singapore, 2023, pp. 2511–2522. URL: https://aclanthology.org/2023.emnlp-main.153/. doi:10.18653/v1/2023.emnlp-main.153.

[37] H. Tan, Z. Guo, Z. Shi, L. Xu, Z. Liu, Y. Feng, X. Li, Y. Wang, L. Shang, Q. Liu, L. Song, ProxyQA: An alternative framework for evaluating long-form text generation with large language models, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL '24, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 6806–6827. URL: https://aclanthology.org/2024.acl-long.368/. doi:10.18653/v1/2024.acl-long.368.

[38] F. Xu, Y. Song, M. Iyyer, E. Choi, A critical evaluation of evaluations for long-form question answering, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL '23, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 3225–3245. URL: https://aclanthology.org/2023.acl-long.181/. doi:10.18653/v1/2023.acl-long.181.

[39] 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, EMNLP '23, Association for Computational Linguistics, Singapore, 2023, pp. 6465–6488. URL: https://aclanthology.org/2023.emnlp-main.398/. doi:10.18653/v1/2023.emnlp-main.398.

[40] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with MT-bench and Chatbot Arena, in: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, NeurIPS '23, 2023. URL: https://openreview.net/forum?id=uccHPGDlao.

[41] Y. Wang, Z. Yu, Z. Zeng, L. Yang, C. Wang, H. Chen, C. Jiang, R. Xie, J. Wang, X. Xie, W. Ye, S. Zhang, Y. Zhang, Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization, in: Proceedings of the International Conference on Learning Representations (ICLR), 2024.

[42] F. Şahinuç, S. Dutta, I. Gurevych, Expert preference-based evaluation of automated related work generation, 2026. URL: https://arxiv.org/abs/2508.07955. arXiv:2508.07955.

[43] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, B. Zhu, H. Zhang, M. I. Jordan, J. E. Gonzalez, I. Stoica, Chatbot Arena: An open platform for evaluating LLMs by human preference, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[44] B. Kargi, D. Salinas, From uncertain judgments to calibrated rankings: Conformal Elo estimation for LLM evaluation, 2026. URL: https://arxiv.org/abs/2606.13221. arXiv:2606.13221.

[45] S. Kim, J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee, M. Seo, Prometheus 2: An open source language model specialized in evaluating other language models, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP '24, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 4334–4353. URL: https://aclanthology.org/2024.emnlp-main.248/. doi:10.18653/v1/2024.emnlp-main.248.

[46] Q. T. Le, I. Badache, M. E. A. Hamri, Outsider at CheckThat! 2026: Retrieval, Verification, and Generation for Multilingual Fact-Checking, in: [55], 2026.

[47] D. Galat, M.-A. Rizoiu, UTS at CheckThat! 2026: Cite-Frame Engineering for Generated Fact-Checking Articles, in: [55], 2026.

[48] H. T. T. Truong, S. Tian, DS@GT ARC at CheckThat! 2026: A Verdict-Anchored Pipeline for Generating Fact-Checking Articles, in: [55], 2026.

[49] R. Svahn, M. Maistro, UCPH-RMT at CheckThat! 2026: Evaluating Text Summaries, Atomic Facts, and Semantic Triplets as RAG Context for Fact-Checking Article Generation, in: [55], 2026.

[50] F. S. Hasan, A. S. Anik, E. Liu, X. Song, M. Rashid, L. Hong, SourceMinds at CheckThat! 2026: NLI-Grounded Citation Auditing in a Multi-Agent Pipeline for Full Fact-Checking Article Generation, in: [55], 2026.

[51] T. Munawar, A. Amir, F. Alvi, A. Samad, Checkmate at checkthat! 2026: Fusion-based retrieval, reasoning-trace ranking, and citation-controlled article generation, in: [55], 2026.

[52] B. Karlash, KNU Fact at CheckThat! 2026: Exploring context engineering and chain-of-thought strategies for fact-checking article generation, in: [55], 2026.
[53] S. M. K. Raza, F. Alvi, A. Samad, AKSH at CheckThat! 2026: A Three-Stage Multi-Agent Pipeline for Automated Fact-Checking Article Generation, in: [55], 2026.
[54] OpenAI, OpenAI GPT-5 system card, 2026. URL: https://arxiv.org/abs/2601.03267. arXiv:2601.03267.
[55] E. S. Salido, A. Barrón-Cedeño, Alba García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CLEF 2026, Jena, Germany, 2026.