Ro at CheckThat! 2026: Evidence-Grounded Base Article Construction with Citation-Locked Rewriting for Full Fact-Checking Article Generation

CheckThat! at CLEF 2026

Jieren Luo[1], Jiacheng Tang[1], Kaichuan Lin[1] and Zhongyuan Han*[1]

[1]Foshan University, Foshan, China

Abstract

CLEF 2026 CheckThat! Task 3 asks systems to generate a complete fact-checking article from a claim, its verdict label, and the provided `premise_articles`. We present a two-stage system for this setting that combines Evidence-Grounded Base Article Construction with Citation-Locked Rewriting. The first stage builds the main article draft by decomposing the claim, selecting and verifying evidence nuggets from the provided sources, and choosing the best citation-audited candidate article. The second stage performs conservative stylistic edits by freezing citation-bearing factual sentences and rewriting only non-cited sentences under sentence-level and document-level guards, while the first stage relies on NLI verification to filter evidence nuggets before article construction. On the official public test leaderboard, our submitted system achieves a mean score of 0.398, ranks sixth, and outperforms the baseline score of 0.272. These results suggest that strong evidence-grounded article construction provides the main performance gain, while citation-locked rewriting provides a modest additional gain over the base article system in citation-faithful fact-checking article generation.

Keywords

Fact-Checking Article Generation, Citation-Locked Rewriting, NLI Verification, Evidence Grounding

1. Introduction

Automatic fact-checking research has traditionally focused on claim detection, evidence retrieval, and verdict prediction. In practical fact-checking workflows, however, the final product is not a label alone but a complete fact-checking article that explains the verdict, organizes supporting and conflicting evidence, and attributes factual statements to concrete sources. CLEF 2026 CheckThat! Task 3 turns this practical requirement into a shared task [1, 2]: given a claim, its verdict label, and a set of `premise_articles`, participants must generate a complete fact-checking article with inline citations to the provided evidence sources.

This setting is challenging because two desirable properties can easily conflict. A system may generate fluent prose that reads like a fact-checking article while attaching weak or unnecessary citations; conversely, a system may stay close to the evidence but produce fragmented or awkward text. The official task uses WatClaimCheck [3] as its underlying dataset. Each instance contains claim metadata, a verdict label, a professional review article, and `premise_articles` used as evidence. This makes citation faithfulness central: a generated article is useful only when its cited sentences remain supported by the cited sources.

Our development experiments suggested that aggressive whole-article rewriting is risky in this setting. When citation-bearing and non-citation sentences are edited together, stylistic gains may come with evidence-aligned content drift, reducing citation precision or citation recall. We therefore treat the task as evidence-grounded, citation-constrained article generation and propose a two-stage pipeline. The first stage, Base Article Construction, produces the main evidence-grounded article

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

*Corresponding author.

wolike666@gmail.com (J. Luo); koppen0529@gmail.com (J. Tang); 1441584221bruan@gmail.com (K. Lin); hanzhongyuan@gmail.com (Z. Han*)

0009-0007-4867-4214 (J. Luo); 0009-0008-7343-8982 (J. Tang); 0009-0001-7843-2888 (K. Lin); 0000-0001-8960-9872 (Z. Han*)

through subclaim generation, passage selection, nugget verification, and citation-audited candidate selection. The second stage, Citation-Locked Rewriting, rewrites only non-cited transition sentences while freezing citation-bearing factual sentences under sentence-level and document-level checks. This design aims to maintain local coherence while strictly preserving citation anchors.

2. Related Work

Automated fact-checking is commonly modeled as a pipeline involving claim interpretation, evidence retrieval, reasoning, and verdict prediction [4, 5, 6, 7]. FEVER and SciFact established canonical benchmarks for claim verification over Wikipedia and scientific abstracts, respectively [4, 5]. Recent work also studies when evidence is insufficient for a reliable verdict, reinforcing the need for conservative evidence checks [8].

More recent work has started to move beyond verdict prediction toward explanation and full-article generation. Sahnan et al. [9] explicitly introduced the task of automatically generating full fact-checking articles and argued that fact-checking systems should support the final communication stage of professional fact-checking, not only internal verdict prediction. CLEF 2026 CheckThat! Task 3 formalizes this direction as a shared-task benchmark [1, 2].

WatClaimCheck [3] is the official source of the task's training and validation data. The dataset provides claims, professional review articles, verdict labels, and premise articles used by fact-checkers. In the original WatClaimCheck work, the main focus was claim entailment and inference from premise articles. In this shared task, the same evidence structure is used for a generation problem: systems must produce article text that is faithful to the supplied `premise_articles`. The broader task setting is also connected to ExClaim, introduced with JustiLM [10], which studies justification generation for real-world fact-checking claims and contributes part of the public test material used in Task 3.

Citation-aware generation is closely related to our design. Prior work has shown that language models may generate fluent text with weak source support, motivating explicit citation-quality controls [11]. In a related end-to-end setting, FactCheck Editor [12] frames fact-checking as evidence-grounded text revision. Our system follows this motivation in a two-stage setting: it first constructs an evidence-grounded base article and then applies a constrained editing stage that locks citation-bearing factual sentences and restricts rewriting to non-cited transition sentences.

3. Method

3.1. Task Formulation

Let a claim package be denoted by $x = \{c, v, P\}$, where c is the claim text, v is the verdict label, and $P = \{p_1, \ldots, p_n\}$ is the set of `premise_articles`. The goal is to generate a fact-checking article y composed of natural-language sentences and inline citations to URLs drawn from P.

Following the official evaluation, a good output should simultaneously maximize entailment score, citation precision, citation recall, and evidence coverage. Our design goal is to maintain article structure and citation faithfulness without introducing unsupported edits.

3.2. Pipeline Overview

Base Article Construction builds the main evidence-grounded article from the claim package by decomposing the claim, selecting and verifying evidence nuggets from `premise_articles`, and choosing the best citation-audited draft. Citation-Locked Rewriting is then applied as a constrained editing layer: citation-bearing factual sentences are treated as fixed anchors, while only non-cited transition sentences are eligible for rewriting. This restriction makes rewriting conservative by construction: edits can adjust transitions and local phrasing, but they cannot directly alter the sentences whose claims are tied to specific source URLs. Figure 1 summarizes this two-stage structure. As shown in the

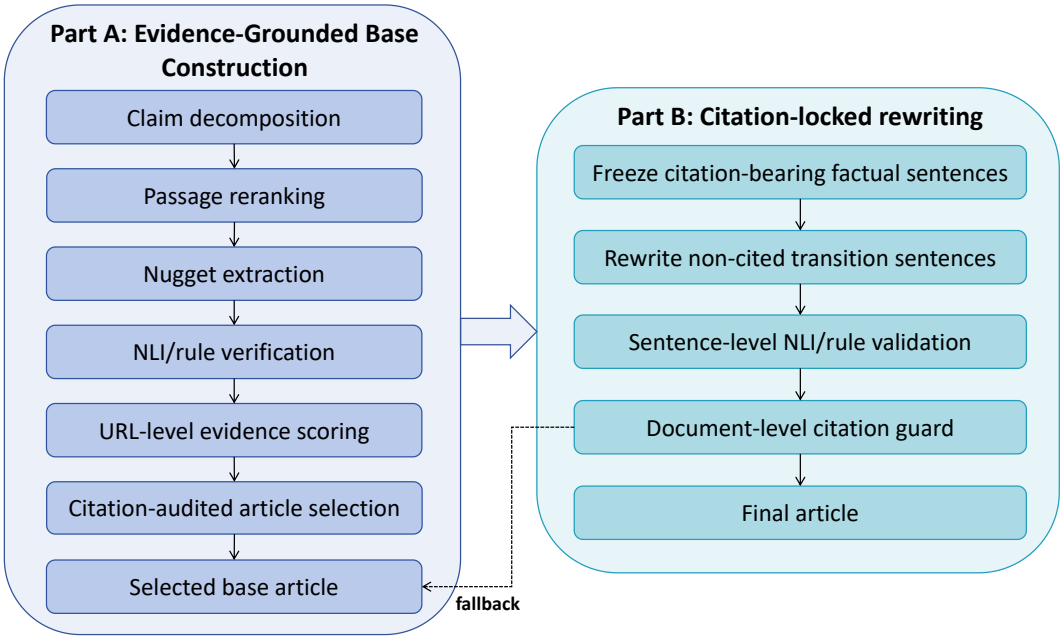

Figure 1: Overall pipeline. Part A corresponds to Base Article Construction, and Part B corresponds to Citation-Locked Rewriting.

ablation results, allowing broader rewriting creates less stable citation behavior than restricting edits to non-cited transition sentences.

3.3. Base Article Construction

The base article is produced by an evidence-grounded construction pipeline that aims to create the main factual draft before any rewriting is attempted. For each claim package, the system first creates a small set of retrieval targets. It prompts `deepseek-v4-flash` to generate atomic subclaims or checking questions from the claim and the supplied `premise_articles`, following prior work on subquestion generation for complex claims [13]. A heuristic decomposition path is retained only as a backup when the model output is unusable.

The evidence documents are then segmented into passages and reranked against these retrieval targets. From the top-ranked passages, the system extracts short evidence nuggets with role labels such as support, contradiction, limitation, or context. Each nugget is verified with our rule-augmented NLI verification mode (`nli_rules`), which combines RoBERTa-large-MNLI entailment probabilities with hard consistency checks over numbers, dates, and named entities. This step is intended to filter passages that are topically related but not sufficiently faithful at the token level. When this main verified-nugget path yields no usable evidence, the implementation backs off to a simpler passage-based extraction procedure so that the system can still return a non-empty article.

The verified nuggets are aggregated by source URL and converted into several candidate articles. Each candidate is then passed through the same citation audit used later in the rewriting stage. This audit records internal support statistics for the candidate, including how many cited sentences remain supported by their cited URLs, how many evidence URLs are covered, and how often a sentence can be supported by a single citation rather than a citation bundle. The final base article is selected by an internal audit-based rule score, that is, a development-time ranking score used only to choose among candidate drafts for the same claim. The score combines these citation-audit statistics with penalties for unsupported sentences and coarse length mismatch. It is therefore a candidate-selection heuristic rather than an official evaluation metric. Overall, the system does not rely on a single generation path. It combines a backup decomposition path, a sparse-evidence recovery path when verified nuggets are

unavailable, and multi-candidate audit-based selection before one base article is chosen. This selected article becomes the input to Citation-Locked Rewriting.

3.4. Citation-Locked Rewriting

The main idea of Citation-Locked Rewriting is that factual citation anchors should not be rewritten once they are already citation-safe. Starting from the selected base article, we split the text into sentences and freeze every sentence that already contains an inline citation marker. Only non-cited sentences are treated as transition candidates.

For each transition candidate, we prompt `deepseek-v4-flash` to perform a conservative stylistic rewrite while preserving meaning. The rewritten sentence is accepted only if it satisfies four checks:

- it must remain a single non-cited sentence
- it must remain a short transition sentence rather than a new evidence-bearing statement
- it must not introduce unseen numbers or month tokens relative to the original sentence or claim
- it must satisfy an NLI preservation check against the original sentence, using 0.44 as an empirical operating threshold

While a rigorous, large-scale hyperparameter search was computationally infeasible during the shared task, preliminary checks on the fixed 20-example validation subset (n20) indicated that the system's core citation metrics remain stable within a narrow band from 0.40 to 0.48. Thus, 0.44 serves as a robust heuristic anchor rather than a globally optimized constant.

If a candidate fails validation, we immediately fall back to the original sentence. After all local rewrites are processed, we run a document-level citation audit. The rewritten article is accepted only when internal citation-audit statistics remain close to those of the selected base article, including unsupported sentence count, audit pass rate, single-citation ratio, and covered URL count. Here, audit pass rate denotes the proportion of cited sentences that remain supported after citation repair:

$$\mathrm{AuditPassRate} = 1 - \frac{N_{\mathrm{unsupported}}}{\max(1, N_{\mathrm{cited}})}. \tag{1}$$

All document-level checks compare the rewritten article with the selected base article for the same claim. Otherwise, the entire article is reverted to the base version.

4. Experiments

4.1. Experimental Setup

4.1.1. Experimental Data

We use the official Task 3 data format. WatClaimCheck contains 26,976 training instances and 3,372 validation instances, each with a claim, verdict metadata, a review article, and `premise_articles`. The released test set contains 1,158 claim packages. Our experiments therefore use two evaluation settings. During system development, we evaluated variants on a fixed 20-example validation subset (n20) of the WatClaimCheck validation set. This subset uses the same 20 validation claim IDs as the baseline evaluation file generated from the baseline script. For the final system, we also report the official public test leaderboard result [14].

4.1.2. Data Processing

Each claim package is converted into a unified internal representation containing the claim text, verdict label, reviewer metadata, and `premise_articles`. Each premise article is stored as an evidence document with its source URL, source file, and raw text. The evidence text is split into fixed-length sentence-based passages so that retrieval, nugget extraction, and later citation auditing operate over comparable evidence units rather than full articles.

At implementation time, the first stage decomposes the claim into a small set of atomic retrieval targets, reranks evidence passages against those targets, and passes the top passages to nugget extraction.

For each selected passage, `deepseek-v4-flash` extracts short evidence nuggets with role labels, and each nugget is verified in `nli_rules` mode. RoBERTa-large-MNLI checks passage-to-nugget support and claim relevance, while hard rules screen for mismatched numbers, dates, and named entities. If no nugget passes verification, the system falls back to a simpler passage-based extraction path so that article construction still has minimal evidence material. Verified nuggets are then aggregated by URL, converted into multiple candidate articles, and filtered by citation audit before the selected base article is passed to Citation-Locked Rewriting.

4.1.3. Evaluation Metrics

We follow the official evaluation scripts and report entailment score, citation precision, citation recall, evidence coverage, and mean score. Citation recall and citation precision follow the citation-quality formulation proposed by Gao et al. [11], adapted by the task scripts to the supplied evidence URLs. The official entailment score is computed bidirectionally between generated and reference article chunks, where each article is chunked into groups of three sentences. For citation recall, the script checks whether the concatenated evidence text of all URLs cited in a generated sentence entails that sentence. For citation precision, the script first requires citation recall to hold, and then tests whether each cited URL is necessary: for multi-source citations, removing any cited URL should make the remaining evidence insufficient to entail the sentence. The final article-level citation recall and citation precision are obtained by averaging these sentence-level decisions over generated sentences that contain inline citations. Evidence coverage measures how many provided evidence URLs are cited in the generated article, relative to the total number of available evidence URLs for that claim. The official mean score is the arithmetic mean of entailment score, citation precision, citation recall, and evidence coverage. In our discussion, citation precision and citation recall are particularly important because the second stage is specifically designed to improve citation stability while preserving the evidence structure inherited from the first stage.

4.1.4. Baseline Methods

We compare against the baseline prompting strategy on the same n20 validation subset. The baseline script loads WatClaimCheck validation items, samples examples with a fixed random seed when n is smaller than the full split, and sends each item to an LLM as a single article-generation prompt. The prompt provides the claim, claimant, original rating, and evidence articles keyed by URL. It instructs the model to act as an expert fact-checker and journalist, to use only the provided evidence excerpts, to write a headline, detailed analysis, and conclusion, and to cite sources inline in the required format `(source: <url>)` or `(source: <url1>; <url2>; ...)`. The baseline does not perform explicit claim decomposition, passage reranking, nugget verification, citation auditing, or fallback repair; it relies on the generator to use the supplied evidence and produce valid citations directly. For Table 1, baseline refers to the DeepSeek-compatible run of the baseline prompt with `deepseek-v4-flash` on the same n20 validation subset.

4.2. Experimental Results and Analysis

Table 1 shows that the full two-stage system improves substantially over the baseline generated by the baseline script, with an absolute gain of 0.1753 in mean score. Most of this gain is already achieved by Base Article Construction, which raises the mean score from 0.3585 to 0.5228. Relative to that strong base system, Citation-Locked Rewriting further improves the mean score from 0.5228 to 0.5338. The gain comes from entailment score, citation precision, and citation recall, while evidence coverage remains lower than the baseline at 0.4446. The lower evidence coverage likely reflects the system's conservative citation behavior. Base Article Construction tends to retain a smaller set of higher-confidence evidence URLs and filter out weak or redundant citations, while Citation-Locked Rewriting preserves this citation

Table 1
Development results on n20 validation using the official evaluation scripts.

Method	Entailment	C-Precision	C-Recall	E-Coverage	Mean score
Baseline	0.2674	0.2168	0.2268	0.7228	0.3585
Base Article Construction	0.3244	0.6611	0.6611	0.4446	0.5228
Citation-Locked Rewriting	**0.3388**	**0.6759**	**0.6759**	0.4446	**0.5338**

Table 2
Representative development ablation results on n20.

Method	Entailment	C-Precision	C-Recall	E-Coverage	Mean score
Broader Style Rewriting	0.3259	0.6272	0.6272	0.4446	0.5062
Sentence-Locked Rewriting	0.3407	0.6185	0.6185	0.4446	0.5056
Sentence-Dedup Post-processing	0.3286	0.6620	0.6620	0.4446	0.5243

Table 3
Empirical stability check of the NLI preservation threshold on the n20 subset.

NLI threshold	Entailment	C-Precision	C-Recall	E-Coverage	Mean score
0.40	0.3507	0.6759	0.6759	0.4446	**0.5368**
0.42	0.3244	0.6759	0.6759	0.4446	0.5302
0.44	0.3350	0.6759	0.6759	0.4446	0.5329
0.46	0.3391	0.6759	0.6759	0.4446	0.5339
0.48	0.3367	0.6759	0.6759	0.4446	0.5333

structure rather than expanding it. As a result, the pipeline improves citation faithfulness more reliably than raw URL coverage.

In addition to these controlled development results, our final submission was evaluated on the official public test leaderboard of CLEF 2026 CheckThat! Task 3 [14]. Our system obtained a mean entailment score of 0.321, citation precision of 0.463, citation recall of 0.463, evidence coverage of 0.343, and a mean score of 0.398, ranking sixth on the public leaderboard. The same leaderboard reports a baseline mean score of 0.272. Although the official test result is lower than our n20 development score, it still supports the main claim of the paper: the final system remains competitive under the task's end-to-end evaluation and improves substantially over the baseline in citation-aware article generation.

The validation and test results together suggest a consistent pattern. What worked well was the system's conservative citation strategy: on the n20 development subset, Citation-Locked Rewriting improved over the base article system, and on the official test leaderboard the largest gains over the baseline appeared in citation precision and citation recall. What did not work as well was broader evidence usage. Our system remained relatively conservative in citing additional URLs, and this limited evidence coverage and prevented the final score from matching the highest-ranked systems on the public leaderboard.

The ablations illustrate the effect of different rewriting scopes clearly. Broader rewriting strategies such as Broader Style Rewriting and Sentence-Locked Rewriting can maintain or increase entailment, but they degrade citation precision and citation recall relative to Citation-Locked Rewriting. Mild post-processing such as Sentence-Dedup Post-processing remains reasonably safe but does not match Citation-Locked Rewriting. These results support our main claim that the most reliable editing space is the set of non-cited transition sentences, not the evidence-bearing sentences themselves.

Table 3 presents an empirical stability check of the NLI threshold. Across values from 0.40 to 0.48, citation precision, citation recall, and evidence coverage remain identical, with only minor fluctuations in entailment score and mean score. We explicitly acknowledge that 0.44 is a heuristic setting based on development-time observations. This check therefore serves only to confirm that the citation-locking

mechanism remains stable within this local range, rather than to claim a globally optimal threshold.

Most controlled development experiments were conducted on the fixed 20-example validation subset, so the n20 comparisons should be interpreted as mechanism-oriented development evidence rather than as a statistically strong estimate of effect size over the full validation distribution. Their main purpose is to verify the direction of the pipeline design rather than to support strong claims about small performance differences.

A second limitation is that the system still relies on backup decomposition, sparse-evidence recovery, and multi-candidate selection when the main evidence-verification path is too sparse. Although this design helps avoid empty or clearly degraded outputs, its weaker passage-based recovery components are still less reliable than the main verified-nugget path. A more robust next step would be to replace these weaker recovery steps with targeted second-pass retrieval or reflective evidence repair.

Finally, Citation-Locked Rewriting is conservative. Since it freezes citation-bearing factual sentences, it cannot repair incorrect evidence choices inherited from Base Article Construction. When the base article selects weak, noisy, or incomplete evidence, the second stage can preserve citation structure but cannot fully recover citation precision, citation recall, or evidence coverage.

5. Conclusions

This paper presented a two-stage pipeline for CLEF 2026 CheckThat! Task 3 that combines Base Article Construction with Citation-Locked Rewriting. We model the task as citation-grounded constrained generation, first build a verified base article from premise evidence, and then apply conservative stylistic edits only to non-cited transition sentences under strict sentence-level and document-level guards.

On the fixed 20-example validation subset (n20), the full system achieves a mean score of 0.5338, and on the official public test leaderboard it achieves a mean score of 0.398 and ranks sixth. The experimental results suggest that strong evidence-grounded article construction provides the main performance gains, while explicit citation locking provides a modest additional gain over the base article system in factual and citation faithfulness. Evidence coverage remains a key direction for future improvement.

Acknowledgments

This work is supported by the National Social Science Foundation of China (24BYY080).

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5.4 for grammar and spelling checks. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. M. Struß, S. Schellhammer, S. Dietze, V. V., V. Setty, T. Chakraborty, P. Nakov, A. Anand, P. Chungkham, S. Hafid, D. Sahnan, K. Todorov, Overview of the CLEF-2026 CheckThat! Lab: Advancing multilingual fact-checking, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer, 2026, pp. 325–335.

[2] D. Sahnan, T. Chakraborty, P. Nakov, Overview of the CLEF-2026 CheckThat! lab task 3 on generating full fact-checking articles, in: E. S. Salido, A. Barrón-Cedeño, A. G. S. de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, Jena, Germany, 2026.

[3] K. Khan, R. Wang, P. Poupart, WatClaimCheck: A new dataset for claim entailment and inference, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), 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. URL: https://aclanthology.org/2022.acl-long.92/. doi:10.18653/v1/2022.acl-long.92.

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

[5] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, H. Hajishirzi, Fact or fiction: Verifying scientific claims, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 7534–7550. URL: https://aclanthology.org/2020.emnlp-main.609/. doi:10.18653/v1/2020.emnlp-main.609.

[6] 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.

[7] X. Zeng, A. S. Abumansour, A. Zubiaga, Automated fact-checking: A survey, Language and Linguistics Compass 15 (2021) e12438. URL: https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/lnc3.12438. doi:10.1111/lnc3.12438.

[8] P. Atanasova, J. G. Simonsen, C. Lioma, I. Augenstein, Fact checking with insufficient evidence, Transactions of the Association for Computational Linguistics 10 (2022) 746–763. URL: https://aclanthology.org/2022.tacl-1.43/. doi:10.1162/tacl_a_00486.

[9] 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?, 2025. URL: https://arxiv.org/abs/2503.17684. arXiv:2503.17684.

[10] 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.

[11] T. Gao, H. Yen, J. Yu, D. Chen, Enabling large language models to generate text with citations, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 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.

[12] V. Setty, Factcheck editor: Multilingual text editor with end-to-end fact-checking, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 2744–2748. URL: https://doi.org/10.1145/3626772.3657663. doi:10.1145/3626772.3657663.

[13] J. Chen, A. Sriram, E. Choi, G. Durrett, Generating literal and implied subquestions to fact-check complex claims, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3495–3516. URL: https://aclanthology.org/2022.emnlp-main.229/. doi:10.18653/v1/2022.emnlp-main.229.

[14] CheckThat! Lab Organizers, Checkthat! lab at clef 2026: Task 3 generating full fact-checking articles, https://checkthat.gitlab.io/clef2026/task3/, 2026. Accessed July 2, 2026.