The Second Bridge on Artificial Intelligence for Scholarly Communication (AAAI-26)
Contributions
https://doi.org/10.52825/ocp.v8i.3161

Published: 22 Dec. 2025

Unmediated AI-Assisted Scholarly Citations

Stefan Szeider

TU Wien, Vienna, Austria

Abstract. Traditional bibliography databases require users to navigate search forms and manually copy citation data. Language models offer an alternative: a natural-language interface where researchers can write text with informal citation fragments and have them automatically resolved to proper references. However, language models generate fabricated (hallucinated) citations at substantial rates, making them unreliable for scholarly work. We present an architectural approach that combines the natural language interface of LLM chatbots with the accuracy of direct database access, implemented through the Model Context Protocol. Our system enables language models to search bibliographic databases, perform fuzzy matching, and export verified entries, all through conversational interaction. A key architectural principle bypasses the language model during final data export by fetching entries directly from authoritative sources, with timeout protection, to guarantee accuracy. We demonstrate this approach with MCP-DBLP, a server providing access to the DBLP computer science bibliography. The system transforms form-based bibliographic services into conversational assistants that maintain scholarly integrity. This architecture is adaptable to other bibliographic databases and scholarly data sources.

Keywords: Language Models, Bibliographic Databases, Model Context Protocol, Citation Management, Scholarly Communication

1. Introduction

1.1 The Opportunity

Bibliographic databases provide authoritative metadata for scholarly publications, but require users to navigate search interfaces, fill in form fields, and manually copy citation data. Consider a researcher writing a manuscript with informal citations like:

```
Transformers have revolutionized NLP (Vaswani et al. 2017), with
subsequent work on BERT (Devlin paper from 2018) and GPT architectures
(Brown et al. neurips'20)...
```

To convert these to proper BibTeX entries, the researcher must open the database website, search for each reference using partial information (“Devlin paper from 2018”, “neurips’20”), identify the correct publication from results, and copy the citation data, repeating this process for every citation.

Language models offer a compelling alternative: a conversational interface where the researcher provides draft text with citation fragments and receives properly formatted references. The language model understands informal citations, searches the database, handles ambiguities through clarification questions, and produces the final bibliography. This approach transforms bibliographic services from form-based tools into interactive assistants that understand research context.

1.2 The Obstacle

Language models generate fabricated citations ("hallucinations"). Agrawal et al. [1] tested GPT-3, ChatGPT, and GPT-4, finding error rates declining from over 70% to under 50% with more advanced models. Kim et al. [2] examined citations from four models, discovering fabrication rates exceeding 80% for certain publication categories. These fabrications include plausible–but nonexistent–author names, venues, and DOIs. Simply connecting a language model to a database API does not solve this problem: the model may still fabricate or corrupt bibliographic data while processing search results.

Several approaches address citation quality. Agrawal et al. [1] proposed indirect-query self-consistency checks that reduce false discovery rates by 20–30%. The ALCE benchmark [3] evaluates citation quality in long-form question answering, showing that even the best systems provide complete supporting citations for only about 50% of answers. Aly et al. [4] use weakly-supervised fine-tuning to improve citation F1 scores by 34.1 points. Ye et al. [5] introduce AGREE, combining tuning with test-time adaptation. Li et al. [6] propose Citation-Enhanced Generation, regenerating unsupported claims until citations back each sentence. Detection methods include RefChecker [7] and CiteFix [8]. Despite these efforts, language models remain unreliable producers of bibliographic data.

1.3 The Solution

We present an architectural approach that preserves the conversational interface while guaranteeing citation accuracy. The key insight is to separate natural-language interaction from data retrieval. Language models handle understanding informal citations, disambiguating references through dialogue, and managing iterative search. Databases provide accurate metadata. By connecting these components through a protocol that keeps authoritative data retrieval outside the language model's generation loop, we achieve both natural interaction and verified accuracy.

We implement this architecture using the Model Context Protocol (MCP) [9], a standardized interface for connecting language models with external tools. The MCP enables stateful interactions where servers provide tools that language models invoke through structured requests. Since its November 2024 release, over 7,000 MCP servers have been developed [10].

Our approach consists of two architectural principles. First, *conversational search through tool calling*: language models invoke search tools with parameters extracted from natural language queries. The tools return structured results that models can present, filter, or use for follow-up searches. Second, *unmediated database export*: the system retrieves BibTeX entries directly from the database, with timeout protection, and writes them to files, bypassing the language model entirely. Only citation key replacement occurs outside the database, using deterministic pattern matching. This ensures that author names, titles, venues, and DOIs come directly from authoritative sources.

We demonstrate this architecture with *MCP-DBLP*, a server providing conversational access to the DBLP computer science bibliography (over 6 million publications). The architectural principles apply to any bibliographic database with a programmatic interface (PubMed, arXiv, Semantic Scholar, or institutional repositories). MCP-DBLP implements eight tools: instructions retrieval, boolean search, fuzzy title and author matching, venue information retrieval, statistical analysis, and a two-step BibTeX export. Evaluation on 104 obfuscated citations (averaged across three experiments) shows 82.7% perfect match rate for MCP-DBLP with unmediated export versus 28.2% for standard web search, with zero cases of metadata corruption.

The system supports both interactive scholarly writing through AI chat applications (Claude Desktop, Cursor, etc.) and autonomous research agents requiring verified citation capabilities. The standardized MCP interface makes it composable with other research tools, enabling multi-agent workflows where citation management integrates with literature analysis, writing assistance, and fact-checking.

MCP-DBLP is available on PyPI (https://pypi.org/project/mcp-dblp/) with source code at https://github.com/szeider/mcp-dblp.[1]

2. Related Work

The Model Context Protocol ecosystem includes numerous servers for research applications. For literature discovery, servers exist for arXiv, PubMed, Semantic Scholar, and OpenAlex (see Appendix A for repository listings). The Scientific-Papers-MCP server aggregates multiple sources including arXiv, OpenAlex, PubMed Central, bioRxiv, and medRxiv. For personal library management, multiple Zotero MCP servers provide vector similarity search, PDF annotation extraction, and cloud synchronization. With its first release in February 2025, MCP-DBLP was among the first bibliography-focused MCP servers.

Most MCP scholarly servers focus on search and metadata retrieval and do not provide citation export. Among those that do, some provide multi-format export (BibTeX, JSON, CSV) or support multiple citation styles (RIS, BibTeX, APA, MLA). However, these servers return formatted citation text in tool responses, meaning the language model receives—and could inadvertently modify—the citation data before presenting it to users. MCP-DBLP implements *unmediated export*: the export tool writes bibliographic data directly to files on disk and returns only file paths to the language model, bypassing the model's context entirely. This architectural choice eliminates citation corruption and reformatting during the export workflow.

3. System Architecture

3.1 Overview

MCP-DBLP implements the Model Context Protocol server interface, enabling any MCP-compatible client to access DBLP through eight tools. The language model automatically selects appropriate tools based on user queries; users interact through natural language without choosing tools directly. The protocol operates as a stateful client-server architecture in which the server maintains session context across multiple

[1] The repository includes installation instructions, an instruction prompt for LLM guidance, and a test suite with 49 automated tests.

tool calls. This design supports iterative workflows where language models can search, refine queries, and export results through sequential tool invocations.

The system is implemented in Python 3.11+ and uses the official MCP SDK. It communicates with clients via standard input/output streams using JSON-RPC 2.0, making it compatible with desktop applications such as Claude Desktop, as well as web-based clients through the Anthropic MCP Connector. Figure 1 shows the system architecture.

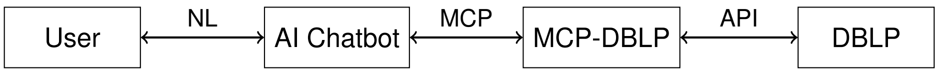

Figure 1. *System workflow: users interact with an AI chatbot, which communicates with MCP-DBLP via the Model Context Protocol. MCP-DBLP queries DBLP via its public API.*

3.2 DBLP API Integration

The DBLP API provides JSON-formatted publication metadata including titles, authors, venues, years, DOIs, and URLs. MCP-DBLP wraps this API with error handling, timeout protection (10 seconds for all requests), and result filtering. When the API times out or returns errors, the system provides informative error messages.

3.3 Tool Implementation

The server exposes eight tools through the MCP interface (Table 1). Search tools accept query strings, year ranges, venue filters, and similarity thresholds (0.0 to 1.0) for fuzzy matching using Python's `difflib.SequenceMatcher`. Statistical tools compute aggregate metrics including publication counts, time ranges, top authors, and top venues.

Table 1. *MCP-DBLP tool specifications*

Tool	**Function**	**Key Parameters**
`get_instructions`	Usage guidance and workflow	(none)
`search`	Boolean search with and/or	query, max results, year range, venue filter
`fuzzy_title_search`	Fuzzy title matching	title, similarity threshold, filters
`get_author_publications`	Author search with fuzzy matching	author name, similarity threshold
`get_venue_info`	Venue metadata retrieval	venue name
`calculate_statistics`	Aggregate statistics	publication list
`add_bibtex_entry`	Add entry to collection	dblp_key, citation_key
`export_bibtex`	Write collection to .bib file	path

3.4 Unmediated BibTeX Export Mechanism

The BibTeX export implements a collection-based workflow that bypasses the language model entirely. The workflow resembles an online shopping cart: the language model calls `add_bibtex_entry(dblp_key, citation_key)` immediately after finding each publication, adding it to a session-scoped collection, and later calls `export_bibtex()` to write all collected entries to a file. This "add to cart" pattern is deliberate: by adding entries immediately after search, the DBLP key is still fresh in the model's context, reducing the risk of key corruption that would occur if the model had to recall keys from earlier in the conversation.

For each `add_bibtex_entry` call, the server constructs the DBLP URL (`https://dblp.org/rec/{dblp_key}.bib`), fetches the BibTeX entry with a 10-second timeout, replaces the citation key using regex pattern matching, and adds the entry to the collection. This design provides immediate verification: if the DBLP key is invalid or the fetch fails, an error is reported back to the language model, which can then retry or inform the user. The "checkout" step, `export_bibtex(path)`, writes the collection to the specified file path and returns the resulting file reference. The bibliographic data (author names with special characters, titles, venues, DOIs) comes directly from DBLP without language model interpretation. The server provides an instruction prompt through the MCP prompt interface explaining tool usage and best practices.

4. Use Cases

The primary use case involves researchers writing papers with AI assistance through applications like Claude Desktop. A researcher can instruct the language model to: (1) identify informal citations in a draft; (2) search DBLP for each reference using fuzzy matching; (3) generate proper citation keys; and (4) export BibTeX entries. The language model uses MCP-DBLP's search and fuzzy_title_search tools to find matching publications, presents candidates for user confirmation, and then calls export_bibtex with the chosen citation keys. The result is a verified .bib file ready for LaTeX compilation.

A related use case involves a researcher who already has citations in BibTeX, but accumulated over time with non-uniform formatting (some conference names abbreviated, some in full, some with date and location, some without). The researcher can use the language model to replace the .bib file with a fresh, certified-correct version in a uniform format. This workflow demonstrates the value of stateful MCP interactions: the language model can iterate through multiple searches, handle ambiguous matches through conversation with the user, and finally export only the confirmed citations.

MCP-DBLP also supports autonomous agents that need citation capabilities. A *deep research* agent tasked with "write a survey of recent work on graph neural networks" can use MCP-DBLP to search for relevant papers, extract key contributions through web search or paper access tools, and generate a properly cited survey with verified references. The MCP protocol's tool-description system and the server's instruction prompt provide the necessary context for language models to use DBLP tools correctly in autonomous workflows.

5. Evaluation

5.1 Methodology

We evaluated bibliography retrieval with and without MCP-DBLP access across 104 obfuscated academic citations across three independent experiments [11]. The baseline (*Web*) used web search only, while MCP-enabled methods used DBLP search with either manual BibTeX construction (*MCP-M*) or direct unmediated export (*MCP-U*).

Ground truth consisted of 104 papers sampled from DBLP, with stratified sampling: 50% from 2020–2025, 25% from 2015–2019, and 25% from 2010–2014. All BibTeX entries were fetched directly from DBLP with 100% success rate. Input citations were obfuscated at varying difficulty levels, ranging from full author name with year to minimal topic hints. Example inputs include "Grassi's paper on computer virus from 2025" and "hybrid algorithm paper on auvs task by Sun 2024."

Each experiment used Claude Code subagents (Claude Sonnet 4.5) with three configurations: Web (no MCP-DBLP), MCP-M (mediated), MCP-U (unmediated). Web relied only on WebSearch and WebFetch tools. MCP-M used MCP-DBLP search tools but constructed BibTeX entries manually, passing citation data through the language model context. MCP-U used the collection-based export API where bibliographic data writes directly to files.

Citations were classified using a 6-category framework: *Perfect Match* (PM) indicates correct retrieval with all core fields correct; *Wrong Paper* (WP) indicates a different paper than ground truth; *Not Found* (NF) indicates missing or failed retrieval; *Incomplete Metadata* (IM) indicates missing doi, pages, volume, or abbreviated names; *Incomplete Author* (IA) indicates truncated author list; *Corrupted Metadata* (CM) indicates wrong values in fields.

5.2 Results and Discussion

Table 2 shows averaged results across the three experiments. The primary finding is that MCP-DBLP eliminates metadata corruption entirely. Both MCP-M and MCP-U achieved CM = 0%, compared to 6.7% for Web. The CM errors in Web are not typos but plausible fabrications: the LLM "corrects" unfamiliar names to common variants (e.g., "Ma'mon Abu Hammad" becomes "Manal Abu Hammad") or invents page numbers when actual values are unavailable.

Table 2. *Averaged results across three experiments (104 citations each). Web = web search only, MCP-M = MCP-DBLP with mediated export, MCP-U = MCP-DBLP with unmediated export.*

Category	Web	MCP-M	MCP-U	Description
PM	28.2%	47.1%	82.7%	Perfect Match
WP	18.6%	15.1%	15.7%	Wrong paper (ambiguous query)
NF	30.1%	1.3%	1.6%	Not found
IM	11.9%	36.5%	0.0%	Incomplete metadata
IA	4.5%	0.0%	0.0%	Incomplete authors
CM	6.7%	0.0%	0.0%	Corrupted metadata

MCP-DBLP also reduces retrieval failures: NF dropped from 30.1% (Web) to 1–2% (MCP-M, MCP-U). The remaining NF errors occur when papers lack DBLP indexing or input citations are too vague. In contrast, WP remained at 15–19% across all methods, indicating these errors stem from citation ambiguity rather than the retrieval method. When "Chaki ieee25" matches multiple DBLP papers, any method may return an unintended paper.

The comparison between MCP-M and MCP-U reveals a trade-off. MCP-M shows 36.5% IM because the agent constructs BibTeX manually, often omitting DOI, volume, or pages. MCP-U achieves 0% IM by exporting directly from DBLP. This validates unmediated export: every MCP-U entry contains exactly the metadata provided by DBLP.

Our experiments used non-interactive mode, where the agent processed citations autonomously without user feedback. In real-world applications, users would typically instruct the language model to ask for clarification when facing ambiguous references. When "Chaki ieee25" matches multiple DBLP papers, the model can present candidates and let the user select the intended one. We expect this interactive disambiguation to reduce both WP and NF rates, as the user can provide additional context or confirm matches that the agent would otherwise guess or skip.

6. Conclusion

We present an architectural approach for connecting language models with bibliographic databases that combines conversational interaction with verified accuracy. The key insight is separating natural language understanding (handled by language models) from data retrieval (handled by databases). MCP-DBLP implements this approach through the Model Context Protocol, providing conversational access to DBLP with unmediated BibTeX export.

Evaluation across three independent experiments with 104 obfuscated citations each shows 82.7% PM for MCP-U versus 28.2% for Web, a 2.9$\times$ improvement, with zero metadata corruption. The results demonstrate that specialized database tools with architectural safeguards can provide both natural language interaction and publication-quality citations. The MCP architecture provides standardization, statefulness, and composability, making the approach broadly applicable across bibliographic databases and research disciplines.

Data availability statement

The MCP-DBLP source code and all implementation details are publicly available at https://github.com/szeider/mcp-dblp under an open-source license.

Author contributions

Stefan Szeider: Conceptualization, Software, Writing (Original Draft, Writing, Review and Editing).

Competing interests

The author declares that he has no competing interests.

Acknowledgements

The author thanks the DBLP team for maintaining the comprehensive computer science bibliography and providing free API access. The research was conducted within Cluster of Excellence *Bilateral Artificial Intelligence* of the Austrian Science Fund (FWF, 10.55776/COE12).

References

[1] A. Agrawal, M. Suzgun, L. Mackey, and A. Kalai, "Do language models know when they're hallucinating references?" In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, Y. Graham and M. Purver, Eds., Association for Computational Linguistics, 2024, pp. 912–928. [Online]. Available: https://aclanthology.org/2024.findings-eacl.62.

[2] E. Kim, F. Kipchumba, and S. Min, "Geographic variation in LLM DOI fabrication: Cross-country analysis of citation accuracy across four large language models," *Publications*,

vol. 13, no. 4, p. 49, Oct. 2025. DOI: 10.3390/publications13040049. [Online]. Available: https://www.mdpi.com/2304-6775/13/4/49.

[3] T. Gao, H. Yen, J. Yu, and 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 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Association for Computational Linguistics, 2023, pp. 6465–6488. DOI: 10.18653/V1/2023.EMNLP-MAIN.398. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.398.

[4] R. Aly, Z. Tang, S. Tan, and G. Karypis, "Learning to generate answers with citations via factual consistency models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds., Association for Computational Linguistics, 2024, pp. 11 876–11 896. DOI: 10.18653/V1/2024.ACL-LONG.641. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.641.

[5] X. Ye, R. Sun, S. Ö. Arik, and T. Pfister, "Effective large language model adaptation for improved grounding and citation generation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, K. Duh, H. Gómez-Adorno, and S. Bethard, Eds., Association for Computational Linguistics, 2024, pp. 6237–6251. DOI: 10.18653/V1/2024.NAACL-LONG.346. [Online]. Available: https://doi.org/10.18653/v1/2024.naacl-long.346.

[6] W. Li, J. Li, W. Ma, and Y. Liu, "Citation-enhanced generation for llm-based chatbots," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds., Association for Computational Linguistics, 2024, pp. 1451–1466. DOI: 10.18653/V1/2024.ACL-LONG.79. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.79.

[7] X. Hu, D. Ru, L. Qiu, *et al.*, "Refchecker: Reference-based fine-grained hallucination checker and benchmark for large language models," *CoRR*, vol. abs/2405.14486, 2024. DOI: 10.48550/ARXIV.2405.14486. arXiv: 2405.14486. [Online]. Available: https://doi.org/10.48550/arXiv.2405.14486.

[8] H. Maheshwari, S. Tenneti, and A. Nakkiran, "Citefix: Enhancing RAG accuracy through post-processing citation correction," *CoRR*, vol. abs/2504.15629, 2025. DOI: 10.48550/ARXIV.2504.15629. arXiv: 2504.15629. [Online]. Available: https://doi.org/10.48550/arXiv.2504.15629.

[9] Anthropic, *Introducing the model context protocol*, Anthropic Blog, Blog post, Nov. 2024. [Online]. Available: https://www.anthropic.com/news/model-context-protocol.

[10] Meetanshi, *MCP statistics april 2025*, Blog post, 2025. [Online]. Available: https://meetanshi.com/blog/mcp-statistics/.

[11] S. Szeider, *Supplementary material for unmediated ai-assisted scholarly citations*, Zenodo, Nov. 2025. DOI: 10.5281/zenodo.17688073. [Online]. Available: https://doi.org/10.5281/zenodo.17688073.