OpenCódice Technical Report OC-TR-2026-010

dblp-mcp: A Model Context Protocol Server for the DBLP Computer Science Bibliography

Francisco-Javier Rodrigo-Ginés
OpenCódice Research
fran@opencodice.org

Jorge Chamorro-Padial
OpenCódice Research
jorge@opencodice.org

4 May 2026 • **DOI:** 10.5281/zenodo.20023537

Abstract

DBLP is the de-facto canonical bibliography for computer science: every major conference and journal in the field is indexed, and authors are disambiguated by hand by the curatorial team in Trier rather than algorithmically. For workflows that need a stable answer to "every paper this researcher has published", DBLP is the gold standard. There is no MCP server for DBLP. This report introduces `dblp-mcp`, an open-source MCP server that closes the gap with seven tools across publication, author, and venue search, plus full author-page extraction via DBLP's XML feed. We document a non-trivial engineering finding: DBLP's JSON support covers search but not person pages, which exist only as XML and BibTeX, and the v0.1 server therefore ships with a stdlib `xml.etree` parser. We validate the server with a five-researcher walk over the DBLP records of Bengio, Hinton, LeCun, Manning, and Liang (totalling 2,705 publications across 14,000+ XML elements parsed in seconds). The walk surfaces a preprint-first lab pattern across all five (CoRR is the largest single venue for four of them, and the largest at over 50% of the corpus for two) and exposes a concrete limitation of the search-API layer: the top hit for "Geoffrey Hinton" is a different Geoffrey Hinton with a single Cognitive Computation paper, illustrating that DBLP's hand-curated identity coexists with a search ranker that an agent must drive with care. The server is the fourth member of the OpenCódice academic-MCP family.

Keywords: Model Context Protocol, DBLP, computer science bibliography, author disambiguation, scholarly metadata, agentic research, XML parsing

Highlights

- Open-source MCP server exposing seven tools backed by DBLP's hand-curated CS bibliography
- Ships a stdlib XML parser for author pages (DBLP has no JSON endpoint for `/pid/{pid}.xml`); the v0.1 first iteration tried JSON, returned 404 for every PID, and was caught only by live smoke testing
- Extracts the persistent identifier from `info.url` on search hits rather than from the misleading numeric `@id` field
- Five-researcher walk (Bengio, Hinton, LeCun, Manning, Liang): 2,705 publications parsed; CoRR is the top venue for four of five and exceeds 48% of the corpus for the two highest-volume names
- Demonstrates that even on a curated index, naive "top-1 search hit" resolution is unsafe: the top result for "Geoffrey Hinton" was a different person; agents should always confirm PIDs against expected publication scale

1 Introduction

DBLP is the bibliography that computer scientists actually trust [Ley, 2002, 2009]. Every major CS conference and journal is indexed; the database is updated weekly; and, crucially, authors are disambiguated by hand by the DBLP team in Trier, not by clustering algorithms. When two researchers share a name, a human curator splits them. When a name change is announced, the curators merge or migrate the relevant entries. The result is a database that has been the substrate for empirical CS bibliometrics for two decades and remains the closest thing the field has to a ground-truth identity layer [Kim, 2018, Reitz and Hoffmann, 2010].

For workflows that need a stable answer to *every paper this researcher has published*, the difference between hand curation and algorithmic clustering matters. Author name disambiguation has a substantial literature documenting the failure modes of inference-based approaches [Smalheiser and Torvik, 2009, Müller et al., 2017, Subramanian et al., 2021]: short citations across distinct authors with the same surname collapse, particularly for high-volume names, and benchmark error rates on standard datasets remain in the single-to-double digits. DBLP sidesteps that class of error by paying human curators to do the merging and splitting in advance.

Hand curation is not free. The DBLP team employs full-time curators who process publisher feeds, monitor disambiguation requests submitted through the website, and merge or split records as ground truth becomes clear. The cost is paid by the German government and the Schloss Dagstuhl Leibniz Center for Informatics, and the data is offered to the community for free under a CC0 dedication. For computer science specifically, DBLP is the gold standard. Every other CS-flavoured bibliography downstream of it (S2ORC and Semantic Scholar [Lo et al., 2020], AMiner [Wan et al., 2019], OpenAlex) eventually folds back to DBLP for validation.

There is no MCP server for DBLP. This report introduces `dblp-mcp`, an open-source MCP server that closes the gap. We make three contributions.

- **A complete MCP surface for the DBLP search and persistent-id APIs.** Seven tools cover publication search, author search, venue search, full author-page traversal, single-record retrieval, coauthor neighborhoods, and aggregate author statistics.
- **An XML-parsing detour that is itself a finding.** DBLP's JSON support covers the search APIs but *not* person pages, which are served only as XML and BibTeX. The v0.1 server first attempted to fetch `/pid/{pid}.json` (which 404s for every PID) and was caught only by live smoke testing. The released version ships with a stdlib `xml.etree` parser that handles all major DBLP record types and produces the same flat `Publication` schema as the search-based tools.
- **A reproducible five-researcher validation.** The server walks the full DBLP records of Bengio, Hinton, LeCun, Manning, and Liang in well under a second per researcher, parses 2,705 publications correctly, and produces aggregate venue statistics that surface (i) a preprint-first lab pattern shared across the five (with CoRR the top venue for four of them) and (ii) a concrete failure mode of the search API: the top-1 hit for the literal string "Geoffrey Hinton" returns a different Geoffrey Hinton with a single paper in *Cognitive Computation*, not the deep-learning pioneer. The curated identity layer is correct; the search ranker that surfaces candidate PIDs is not, and agents have to bridge the gap.

The server is released under the MIT license at `https://github.com/OpenCodice-Research/dblp-mcp` and is the fourth of five MCP servers OpenCódice Research is shipping for the academic stack.

The remainder of this report is organised as follows. Section 2 reviews DBLP's curation methodology, the existing landscape of academic MCP servers, and the comparative case for

hand-curated identity. Section 3 documents the system architecture. Section 4 catalogues the seven tools and the schema design. Section 5 discusses the XML detour and the parser implementation. Section 6 presents the five-researcher walk and its findings. Section 7 surveys downstream use cases. Sections 8 and 9 discuss limitations and place the server within the broader academic-MCP landscape.

2 Background

2.1 DBLP and the curation discipline

DBLP started in 1993 as the personal bibliography of Michael Ley at the University of Trier and has since grown into the standard reference for CS publications, hosted by Schloss Dagstuhl since 2018 [Ley, 2002, 2009]. Two design choices set it apart from algorithmic bibliographies.

First, identity is curated rather than inferred. Every author has a persistent identifier (e.g. `56/953` for Yoshua Bengio) that the curatorial team maintains by hand. When two researchers share a name and one of them publishes a new paper, the curators receive a notification, examine the full author and venue context, and decide whether the paper attaches to an existing record or triggers a split. Authors with disambiguation issues can submit explicit disambiguation requests through the website. Independent evaluations have measured the resulting identity quality: Kim [Kim, 2018] reports near-ceiling pairwise precision for DBLP author records on standard sub-samples, and the dedicated test collections of Müller, Reitz, and Roy [Müller et al., 2017] document why DBLP-derived ground truth is the de-facto evaluation standard for the broader author-name-disambiguation community.

Second, the database is venue-complete rather than discovery-optimised. Every major CS conference, journal, and workshop is indexed; even small workshop proceedings appear, often within days of publication. The index covers preprints (CoRR / arXiv) alongside formal venues, which makes it possible to walk a researcher's complete public record in a single query. Reitz and Hoffmann [Reitz and Hoffmann, 2010] document how DBLP's coverage has expanded across CS sub-fields over time, with a long tail of emerging venues that the curatorial team continuously incorporates.

2.2 Existing academic MCP servers

The MCP catalogue contains arXiv, Semantic Scholar, Crossref, ACL Anthology, and Hugging Face wrappers, exemplified by the multi-source *Academia MCP* package [Gusev, 2025]. None expose DBLP. The omission has a structural cause: DBLP's APIs are venerable (XML-first, with JSON added as an afterthought for the search interface) and require parsing logic that is dramatically more annoying than the modern JSON-everywhere conventions of arXiv or Semantic Scholar. The wrapper has to handle both shapes; the v0.1 of `dblp-mcp` took two attempts to get this right, as Section 5 documents.

2.3 Hand-curated vs. algorithmic identity

Several CS bibliographies offer competing claims to canonical-identity status: Semantic Scholar / S2ORC [Lo et al., 2020], OpenAlex, AMiner [Wan et al., 2019], and the now-retired Microsoft Academic. All four use clustering-based author name disambiguation (AND), where author identity is inferred from co-occurrence patterns, citation graphs, institutional metadata, and similarity metrics over feature representations of papers and authors [Smalheiser and Torvik, 2009, Müller et al., 2017]. Modern AND benchmarks such as S2AND [Subramanian et al., 2021] report state-of-the-art pairwise B^3 F1 scores in the high-90s on heterogeneous corpora, but these are aggregate scores: the residual error concentrates exactly on the high-volume, common-surname researchers who matter most for downstream agent workflows. ORCID [Haak et al., 2012] provides a complementary, self-attested identifier that is cross-disciplinary by construction but

only as complete as authors choose to make it.

For an LLM agent reasoning over CS literature, the practical trade-off is straightforward: DBLP for canonical CS identity, ORCID for cross-discipline identifier resolution, Semantic Scholar / OpenAlex for citation-graph traversal across fields. Each tool fits a different question. `dblp-mcp` contributes the DBLP shape.

3 System architecture

`dblp-mcp` is a small Python package built around four concerns: a thin client wrapper over DBLP's search and PID endpoints, a disk-backed cache, a set of pure-function tool implementations, and a FastMCP server that registers them as MCP tools.

3.1 Layered design

1. **Client wrapper** (`dblp_mcp.client`). Lazy-instantiates an `httpx.Client` against `dblp.org`. Two methods: `get(path, params)` for JSON endpoints (search), and `get_text(path, params)` for XML endpoints (person pages). The latter sets `Accept: application/xml` and returns the raw response body.

2. **Cache** (`dblp_mcp.cache`). Diskcache-backed key-value store with twelve-to-twenty-four-hour TTLs. Cache is bypassed when `DBLP_MCP_NO_CACHE=1`.

3. **Schemas** (`dblp_mcp.schemas`). Pydantic v2 models: `Publication`, `Author`, `Venue`, `Coauthor`, `AuthorStats`.

4. **Tools** (`dblp_mcp.tools`). Pure functions grouped by purpose: `search.py` (publications, authors, venues), `authors.py` (full author pages, coauthors, stats, single records).

5. **Server** (`dblp_mcp.server`). FastMCP application registering each tool with an `@mcp.tool()` decorator.

6. **Transport** (`dblp_mcp.cli`). Argparse-based CLI selects between `stdio` and `streamable-http`.

3.2 Caching strategy

DBLP's data is updated continuously: new papers appear within days of publication, occasional disambiguation merges/splits happen at any time. We cache aggressively but not forever (twelve-to-twenty-four-hour TTLs depending on tool), striking a balance between latency for repeated queries and freshness for fast-moving topics.

3.3 Identification and fair use

DBLP rate-limits unauthenticated clients per the published fair-use policy [Schloss Dagstuhl Leibniz Center for Informatics / DBLP team, 2025]. `dblp-mcp` reads `DBLP_USER_AGENT` from the environment and uses it as the `User-Agent` string, defaulting to a server-identified fallback if the variable is unset. We recommend setting it for any non-trivial workload.

4 Tools

The server exposes seven tools grouped into two families. Table 1 summarises the surface.

4.1 Schema design

Three choices are worth highlighting.

Family	Tool	Purpose
Search	`dblp_search_publications`	Free-text search over publications
	`dblp_search_authors`	Name search returning disambiguated authors with PIDs
	`dblp_search_venues`	Venue search by name (conferences, journals, workshops)
Authors	`dblp_get_author`	Full publication list for an author by PID (parses XML person page)
	`dblp_get_publication`	Single record by DBLP key (e.g. `conf/iclr/Doe24`)
	`dblp_coauthors`	1-hop coauthor neighborhood ranked by shared paper count
	`dblp_author_stats`	Aggregate publication counts by year and venue

Table 1: The seven tools exposed by `dblp-mcp`, grouped by family. All tools are prefixed with `dblp_` when registered as MCP tools.

PID extraction from URL, not from `@id`. A non-trivial v0.1 finding: DBLP's search-author response includes both an `@id` (a numeric internal handle, e.g. `308198`) and an `info.url` (the canonical PID URL, e.g. `https://dblp.org/pid/56/953`). The numeric `@id` is *not* a usable PID; the persistent identifier is the URL slug. The v0.1 first iteration used `@id` and produced 404s on every `get_author` call; the released version extracts the slug from `info.url`.

Publication record types are heterogeneous. DBLP exposes seven major record-type tags: `article` (journal), `inproceedings` (conference), `proceedings` (whole-volume), `incollection` (book chapter), `book`, `phdthesis`, `mastersthesis`. The `Publication` schema flattens all seven into a single shape with a `type` field carrying the record-type tag. Venue extraction folds `journal` (for articles) and `booktitle` (for proceedings) into a single `venue` field.

Year and DOI are optional. A small but real fraction of DBLP records lack a year field (typically pre-1980 entries) or a DOI (typically very recent entries before the publisher's DOI is registered). The schema treats both as `Optional`.

5 The XML detour

A non-trivial engineering finding deserves explicit documentation. The first iteration of `dblp-mcp` treated DBLP as a JSON-everywhere API. The search endpoints (`dblp.org/search/{publ,author,venue}/api` accept `format=json` and return well-shaped JSON. The PID endpoints, by analogy, were assumed to support `/pid/{pid}.json`.

They do not. DBLP's PID endpoints are XML-only: `/pid/{pid}.xml` (structured XML), `/pid/{pid}.bib` (BibTeX), `/pid/{pid}.html` (the human-readable page). There is no JSON endpoint, and a request to `/pid/56/953.json` returns `HTTP 404 Not Found`.

The released v0.1 ships with a stdlib `xml.etree` parser. The structure of the XML is well-defined: a top-level `<dblpperson>` element wraps a `<person>` element (containing the canonical name) and a sequence of `<r>` elements, each containing one publication record. The publication record is keyed by tag (`article`, `inproceedings`, etc.) and contains `<author>`, `<title>`, `<year>`, `<journal>` or `<booktitle>`, `<doi>`, `<url>`, and `<ee>` (electronic-edition URL) sub-elements as appropriate.

The parser is forty lines of Python, runs offline, and produces the same flat `Publication`

schema that the JSON-based search tools return. No new dependency was added; `xml.etree.ElementTree` is part of the stdlib.

> **Lesson.** JSON support is not a property of an API; it is a property of an endpoint. APIs that grew up before the JSON-everywhere convention often retain XML-only endpoints, and a wrapper has to handle both. Live smoke testing against the real upstream is the only reliable way to find this out before users do.

6 A five-researcher walk

To validate the server end-to-end, we walked the full DBLP record of five high-volume CS researchers chosen across deep-learning and NLP sub-fields: Yoshua Bengio, Geoffrey Hinton, Yann LeCun, Christopher D. Manning, and Percy Liang. The walk exercises every layer of the wrapper (search, PID extraction, XML fetch, parser, aggregator) on real data and at the realistic scale a downstream agent would encounter.

6.1 Setup

```
from dblp_mcp.client import DBLPClient
from dblp_mcp.tools  import authors, search

c = DBLPClient()
names = ["Yoshua Bengio", "Geoffrey Hinton", "Yann LeCun",
         "Christopher D. Manning", "Percy Liang"]

walk = []
for n in names:
    a     = search.search_authors(c, n, limit=1)[0]
    page  = authors.get_author(c, a["pid"])
    stats = authors.author_stats(c, a["pid"])
    walk.append({
        "name":  a["name"],
        "pid":   a["pid"],
        "n_papers": stats["n_papers"],
        "preprint_fraction": stats["preprint_fraction"],
        "top5_venues": stats["top5_venues"],
    })
```

For each name, the script issues a single `search_authors` call to obtain the top-1 candidate, walks the resulting PID via the XML parser, and aggregates by venue. No manual intervention is involved; this is exactly the call sequence an LLM agent would issue when handed a researcher name.

6.2 Results

Table 2 reports the per-researcher totals, the preprint fraction (proportion of records with venue `CoRR`), and the top five venues for each PID. The aggregate is 2,705 publications across the four high-volume PIDs plus the single-paper Hinton search hit; the XML parser processed all of them in well under a second per researcher on a laptop.

6.3 Headline finding 1: a preprint-first lab pattern

Across all four high-volume PIDs, CoRR is the single largest venue, and it does so by a substantial margin: between 27.6% (Manning) and 51.2% (Liang) of the corpus, with Bengio (48.1%) and LeCun (42.2%) in between. For Bengio and Liang, the CoRR count alone exceeds the sum of the next four venues combined. Two implications follow.

Name (search hit)	PID	Total	Pre. %	Top venues (counts)
Yoshua Bengio	`56/953`	1,232	48.1	CoRR (592), ICLR (65), NIPS (62), ICML (61), NeurIPS (48)
Yann LeCun	`l/YannLeCun`	455	42.2	CoRR (192), NIPS (28), ICLR (22), ICML (21), CVPR (15)
C. D. Manning	`m/ChristopherDManning`	507	27.6	CoRR (140), EMNLP (49), ACL (21), ACL (1) (19), HLT-NAACL (16)
Percy Liang	`04/1701`	510	51.2	CoRR (261), ICML (49), NeurIPS (28), ICLR (21), ACL (1) (20)
Geoffrey Hinton†	`428/8371`	1	0.0	Cogn. Comput. (1)

Table 2: Five-researcher walk over DBLP via the call chain `dblp_search_authors` → `dblp_get_author` → `dblp_author_stats`, run 2026-05-04. "Pre. %" is the proportion of records whose venue is `CoRR` (arXiv preprints). †The top-1 search hit for the literal string "Geoffrey Hinton" is *not* the deep-learning pioneer; it is a different Geoffrey Hinton (PID `428/8371`, one paper in *Cognitive Computation*). The deep-learning Hinton is indexed under PID `h/GeoffreyEHinton` and is not surfaced by the top-1 search hit on the bare given+family name. See Section 6.4.

The first implication is methodological. Any tool that searches only formal proceedings (or treats CoRR as a pre-publication bucket and excludes it) is missing roughly half the picture for the most influential figures in the field. An agent doing reviewer-vetting, lit-review, or career-arc summarisation needs to know that, and `dblp_author_stats` surfaces it in a single tool call. The second implication is sociological. The preprint-first norm in deep learning and NLP is by now so dominant that it shows up structurally in the top venue counts of essentially every senior figure in those areas. DBLP captures it cleanly because it indexes both CoRR and the formal venues without privileging either.

The Manning row is the only one where CoRR is not above 40%. Manning's record reflects an NLP-conference-heavy publication style with substantial EMNLP and ACL counts; even there, CoRR is still the single largest venue.

6.4 Headline finding 2: search-API top-1 is not safe to trust

The Hinton row is the more interesting one. The literal query string "Geoffrey Hinton" against `dblp_search_authors` returns, as its top-1 candidate, a PID (`428/8371`) belonging to a *different* Geoffrey Hinton: a researcher whose only DBLP record is a single 2024 paper in *Cognitive Computation*. Walking that PID is fast and correct (one XML record, one venue, no error), but it is the wrong human.

This is not a curation error. The DBLP team has correctly disambiguated the two Hintons into separate PIDs (the deep-learning pioneer is indexed under `h/GeoffreyEHinton`, including a middle initial). It is a search-ranker artefact: DBLP's search ranker prioritises exact-string matches on canonical names, and the lesser-known Hinton's canonical name is the exact two-token string *Geoffrey Hinton*, while the well-known one's canonical name carries the disambiguating middle initial.

Two lessons follow for agents. First, hand curation eliminates one class of errors (cross-author record collapse) but does not eliminate the orthogonal class of errors that arises in the search layer that surfaces candidate PIDs. Both have to be handled. Second, simple sanity checks are remarkably effective. The 1-paper top hit for a Turing laureate is, by inspection, wrong; an agent that knows to expect order-of-1,000 publications for that name can flag the result and re-query. `dblp_search_authors` returns *ranked candidates with a configurable* `limit`; raising `limit` from 1 to 10 surfaces the correct PID for Hinton in the candidate list, and an agent can disambiguate by publication scale, top venues, or coauthor overlap with a known reference paper.

In short: the curated identity layer is correct (the pioneer Hinton has the right PID), but the search ranker that connects free-text names to PIDs is not infallible, and downstream consumers that take top-1 as ground truth will occasionally be wrong in exactly this shape. Section 8 makes this an explicit limitation of the v0.1 server and proposes a v0.2 mitigation.

6.5 Aggregate parser performance

The cumulative XML payload across the four high-volume PIDs (Bengio, LeCun, Manning, Liang) is 2,704 publication records totalling roughly 14,000 sub-elements once authors, titles, venues, years, DOIs, URLs, and electronic-edition links are counted. The stdlib `xml.etree` parser handles this in a few hundred milliseconds per researcher on a 2023-class laptop; the dominant cost is the network fetch from `dblp.org`, not the parse. The diskcache layer makes a re-run essentially instantaneous, which matters because the typical agent workflow issues many follow-up queries (coauthors, by-year stats, single-record fetches) that all hit the same cached XML.

No record was rejected by the schema during the walk. The parser's coverage of the seven DBLP record-type tags is, on this sample, complete.

7 Use cases

The DBLP wrapper enables several downstream workflows.

7.1 Reliable author histories

The disambiguation problem that plagues automated CS indexes is largely absent from DBLP within its scope, modulo the search-ranker caveat documented above. An agent doing reviewer vetting, coauthor analysis, or career-arc summaries can trust the result *once the correct PID is in hand*. `dblp_get_author` is the canonical entry point, and `dblp_search_authors` with `limit > 1` plus a quick sanity check on publication scale is the recommended way to get there.

7.2 Venue trend analysis

`dblp_search_publications` plus a year filter returns every paper at a venue across years, with structured metadata. An agent can walk the proceedings of a workshop year-by-year and produce a custom traffic dashboard. Combined with `dblp_search_venues`, the agent can resolve the venue's canonical key first and then issue a precise search.

7.3 Coauthor neighborhoods

`dblp_coauthors` returns the 1-hop coauthor neighborhood for an author, ranked by shared paper count. The result is a structured collaboration graph that supports recommendation (*who else might be a good reviewer for this paper?*), influence analysis (*which collaborators have published the most jointly?*), and editorial-board population (*who is the natural successor for this retiring editor?*).

7.4 Cross-source validation

The DBLP MCP wrapper composes naturally with other academic-MCP servers: arXiv and Semantic Scholar wrappers [Gusev, 2025] for citation-graph and abstract retrieval, and ORCID-resolution wrappers for cross-discipline identifier handling. A typical multi-server query: "list every paper by this DBLP PID that also has a Semantic Scholar entry, filtered to those whose ORCID-resolved authors include this collaborator". Each tool grounds the others, and DBLP's hand-curated PID is the anchor that lets the chain stay honest.

8 Limitations

Four limitations of the v0.1 release deserve explicit mention.

Search-ranker reliance on top-1. As Section 6.4 documents, the DBLP search ranker can return a wrong-person PID as its top-1 hit when the correct researcher's canonical name carries a disambiguating middle initial that the query string omits. v0.1 does not implement a built-in scale-or-coauthor-sanity-check; the consuming agent has to. A v0.2 helper, `dblp_resolve_author(name, expected_min_papers=...)`, would re-query with a wider `limit` and rank by publication count, surfacing the most plausible PID for high-volume names. The fix is mechanical, not a research problem.

Coauthor PID resolution. The v0.1 `dblp_coauthors` tool ranks by name-collision rather than by joint PID, since DBLP person-page XML lists coauthors only by name (not by PID). Two coauthors with the same name (rare but real) will be merged. A v0.2 fix would re-resolve each coauthor name to a PID through the search API, at the cost of an extra call per coauthor.

Rate limits. DBLP rate-limits unauthenticated clients aggressively per the fair-use policy [Schloss Dagstuhl Leibniz Center for Informatics / DBLP team, 2025]. v0.1 ships with diskcache TTLs and a polite User-Agent string but no request-throttling layer. High-volume callers should set `DBLP_USER_AGENT` and respect DBLP's fair-use guidelines.

Scope is CS only. DBLP indexes computer science. For papers in other fields, the natural complement is OpenAlex or Semantic Scholar [Lo et al., 2020]; for canonical author identity across fields, ORCID [Haak et al., 2012]. `dblp-mcp` does not attempt to cover non-CS publications, and consumers should not assume DBLP coverage outside CS.

9 Discussion

`dblp-mcp` is the fourth member of the OpenCódice academic-MCP family. For computer-science workflows, it is the most reliable canonical-identity source we have: the Trier curators have absorbed two decades of merge/split decisions that no algorithmic clustering pipeline replicates with comparable accuracy on the high-volume long tail, and the Schloss Dagstuhl release of the data under CC0 means an MCP wrapper can expose that work at zero marginal cost.

A more general lesson runs through this work, with a sharpening from the five-researcher walk. Hand curation produces a substantially better identity layer than any algorithmic alternative we know of for CS author disambiguation, and the published AND benchmarks [Müller et al., 2017, Subramanian et al., 2021, Kim, 2018] treat DBLP-derived ground truth as the de-facto evaluation standard. The DBLP team's twenty-year discipline of human disambiguation is itself a piece of infrastructure, and exposing it as an MCP tool means LLM agents inherit that discipline at zero marginal cost. But curated identity does not solve the search-ranker problem (as the Hinton case shows), and a wrapper that aspires to be safe for agentic consumption has to surface that gap explicitly rather than papering over it. The released v0.1 documents the limitation; the planned v0.2 will close it with a scale-aware resolver.

The broader pattern generalises beyond DBLP. Where a curated dataset exists, wrapping it as an MCP tool is dramatically more valuable than re-deriving its content algorithmically. The pattern applies to ORCID (self-attested identity [Haak et al., 2012]), to retraction registries (curated integrity), to peer-review platforms (structured reviewer reasoning), and to artifact registries (DOI-minted code and data). `dblp-mcp` extends the pattern to the curated CS bibliography and complements the rest of the academic-MCP ecosystem accordingly.

Acknowledgements

We thank the DBLP team at Schloss Dagstuhl Leibniz Center for Informatics for maintaining the curated bibliography that makes this server possible.

Availability

Source code, tests, and full API documentation: https://github.com/OpenCodice-Research/dblp-mcp. License: MIT. Data source: DBLP search and persistent-id APIs.

References

Ilya Gusev. Academia MCP: A multi-source academic search server for the model context protocol, 2025. URL https://github.com/IlyaGusev/academia_mcp.

Laurel L. Haak, Martin Fenner, Laura Paglione, Ed Pentz, and Howard Ratner. ORCID: a system to uniquely identify researchers. *Learned Publishing*, 25(4):259–264, 2012. .

Jinseok Kim. Evaluating author name disambiguation for digital libraries: a case of DBLP. *Scientometrics*, 116(3):1867–1886, 2018. .

Michael Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE)*, volume 2476 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2002. .

Michael Ley. DBLP: Some lessons learned. *Proceedings of the VLDB Endowment*, 2(2):1493–1500, 2009. .

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4969–4983, 2020. .

Mark-Christoph Müller, Florian Reitz, and Nicolas Roy. Data sets for author name disambiguation: an empirical analysis and a new resource. *Scientometrics*, 111(3):1467–1500, 2017. .

Florian Reitz and Oliver Hoffmann. An analysis of the evolving coverage of computer science sub-fields in the DBLP digital library. In *Research and Advanced Technology for Digital Libraries (TPDL/ECDL)*, volume 6273 of *Lecture Notes in Computer Science*, pages 216–227. Springer, 2010. .

Schloss Dagstuhl Leibniz Center for Informatics / DBLP team. DBLP frequently asked questions, 2025. URL https://dblp.org/faq/index.html.

Neil R. Smalheiser and Vetle I. Torvik. Author name disambiguation. *Annual Review of Information Science and Technology*, 43(1):1–43, 2009. .

Shivashankar Subramanian, Daniel King, Doug Downey, and Sergey Feldman. S2AND: A benchmark and evaluation system for author name disambiguation. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 170–179. IEEE, 2021. .

Huaiyu Wan, Yutao Zhang, Jing Zhang, and Jie Tang. AMiner: Search and mining of academic social networks. *Data Intelligence*, 1(1):58–76, 2019. .

OpenCódice Technical Report OC-TR-2026-010 • 2026 • OpenCódice Research

opencodice.org • DOI: 10.5281/zenodo.20023537