NUANSA INFORMATIKA
Volume 20 Nomor 1, Januari 2026

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

Operationalizing Model Context Protocol for Multi-Platform Academic Search: A Comparative Study of LLM Grounding

Rizqullah Aryaputra Piliang[1], Anindito[2], Eryan Ahmad Firdaus[3]
[1,2,3]Universitas Pertahanan, Indonesia
E-mail: *[1]**rizqullah.piliang@tm.idu.ac.id**, [2]anindito@idu.ac.id, [3]eryan.firdaus@idu.ac.id

Abstract

This study introduces the Paper Search Model Context Protocol (MCP) server, a unified architecture enabling Large Language Models (LLMs) to autonomously search and retrieve academic literature across heterogeneous platforms (arXiv, PubMed, bioRxiv, medRxiv, Semantic Scholar, and CrossRef). By normalizing diverse metadata into a standardized schema, the system facilitates platform-agnostic tool use. We evaluate this infrastructure by tasking GPT-4.1 and GPT-5.1 with generating grounded literature reviews on AI applications in medicine, cybersecurity, and transportation, measuring performance against retrieved abstracts using ROUGE and BERTScore. Results indicate that GPT-4.1 achieves superior semantic and structural alignment, particularly when grounded via CrossRef (BERTScore F1 = 0.881, ROUGE-L F1 = 0.375), outperforming GPT-5.1 across most metrics. While GPT-5.1 demonstrates higher unigram recall (ROUGE-1 = 0.412 on arXiv), it exhibits lower structural fidelity. These findings validate MCP as a robust integration layer for academic RAG systems and demonstrate that standardized tool interfaces enable precise, quantitative assessment of LLM grounding capabilities.

Keywords— *Model Context Protocol (MCP), Academic paper search, Large language model integration, Retrieval-augmented generation (RAG), Preprint and metadata aggregation*

Submission: November 08, 2025 *Accepted: December 14, 2025* *Published: January 20, 2026*

DOI Article: https://doi.org/ 10.25134/ilkom.v20i1.513

1. INTRODUCTION

The rapid expansion of scientific publishing has intensified the difficulty of performing comprehensive, high-quality literature reviews across multiple digital libraries and indexing services. Researchers increasingly depend on AI-powered academic search tools that provide semantic search, recommendation, and conversational interfaces to navigate large, heterogeneous corpora. Existing systems such as Semantic Scholar, PubMed, and CrossRef offer strong coverage and robust search capabilities, but they generally expose platform-specific APIs or web interfaces rather than a unified, tool-oriented protocol for Large Language Models (LLMs). As a result, contemporary retrieval-augmented generation (RAG) pipelines are often tightly coupled to a single source, a bespoke vector database, or an application-specific integration layer, which complicates reuse, interoperability, and systematic evaluation.

Recent work on academic RAG systems has demonstrated that coupling LLMs with domain-specific corpora can substantially improve factuality, recall, and user satisfaction in scientific question answering and literature exploration [1], [3]. For instance, Yadav and Gopinathan [23] propose a monolithic RAG pipeline using ColBERT and ChromaDB to enhance literature analysis, while Lund [24] highlights the prospects of RAG for academic libraries. Biomedical RAG pipelines built over PubMed or institutional repositories employ dense and sparse retrieval, query expansion, and LLM-based evaluators to support evidence-grounded responses [2], [3]. However, these systems typically implement custom adapters for each upstream data source. At the infrastructure level, large-scale metadata providers such as the Semantic Scholar Academic Graph and CrossRef supply high-quality bibliographic records [4], [5], but they do not by themselves specify how LLMs should invoke tools or exchange typed context in a standardized manner. Regional studies on information systems and AI-enabled decision support likewise highlight the importance of robust data integration and performance evaluation [9],[11].

In parallel, the Model Context Protocol (MCP) has emerged as a general-purpose standard for connecting LLMs to external tools and data sources through a structured JSON-RPC interface [6], [7]. MCP introduces primitives for tools, resources, prompts, and sampling, enabling LLM clients to invoke capabilities in a uniform way. Prior MCP-related studies focus primarily on interoperability, security, and multi-agent coordination [6], [8]. Nevertheless, a critical technical gap remains in the architectural standardization of these systems. Unlike the monolithic pipelines described in [23], current implementations suffer from "adapter fatigue", relying on bespoke, hardcoded adapters for each provider. There is limited work that applies MCP to the specific problem of multi-platform academic search, nor that evaluates how MCP-mediated retrieval affects the lexical and semantic faithfulness of LLM-generated literature reviews.

This paper addresses this gap by introducing and empirically evaluating the Paper Search MCP server, a domain-specific MCP implementation for academic paper search and retrieval. The server exposes standardized search, download, and read tools over heterogeneous scholarly platforms: arXiv, PubMed, bioRxiv, medRxiv, Semantic Scholar, and CrossRef, and normalizes their outputs into a unified Paper representation consumable by LLM clients. Unlike existing academic search systems and RAG stacks that integrate individual APIs directly into application logic or vector databases, Paper Search MCP operationalizes MCP primitives as a cross-platform abstraction layer. This enables a single LLM client to issue uniform tool calls across multiple sources, reason over typed Paper objects, and generate grounded literature reviews while preserving clear provenance.

The main contributions of this work are:

- Unified MCP Architecture: We present a concrete MCP-based architecture for multi-platform academic search that encapsulates platform-specific modules as tools and normalizes responses into a common Paper schema.
- Metric-Driven Evaluation: We propose a methodology combining Lexical ROUGE-N/L and Semantic BERTScore F1 to evaluate LLM-generated reviews against retrieved abstracts.
- Comparative Analysis: We apply this methodology to compare GPT-4.1 and GPT-5.1, demonstrating that GPT-4.1 grounded via CrossRef achieves higher alignment than GPT-5.1, positioning Paper Search MCP as a bridge between protocol-level research and application-level RAG systems.

2. THEORITICAL FRAMEWORK

2.1.1. Model Context Protocol

The Model Context Protocol (MCP) represents a paradigm shift in how Large Language Models (LLMs) interact with external environments. Defined as a standardized specification for connecting AI models to data and tools, MCP abstracts the complexity of API integrations into a uniform JSON-RPC interface [12]. Recent studies have highlighted MCP's role in creating modular, secure, and interoperable AI ecosystems. For instance, Xing et al. [12] propose defense frameworks for MCP integrity, underscoring its growing adoption as the critical "connective tissue" between models and sensitive data sources. Similarly, Song et al. [13] identify that while MCP simplifies tool exposure, allowing developers to define "resources" and "tools" once and have them consumed by any MCP-compliant client, it also necessitates robust security considerations. In the context of this research, MCP serves as the foundational integration layer. Rather than building bespoke connectors for each academic repository (e.g., arXiv, PubMed), the Paper Search MCP server implements the protocol to expose a consistent set of capabilities (search, download, read) that the LLM can autonomously discover and invoke. This aligns with findings by da Silva et al. [14], who demonstrated MCP's utility in manufacturing for exposing diverse functional capabilities to LLM agents without rigid semantic modeling. By leveraging MCP, our system decouples the LLM's reasoning process from the underlying data retrieval mechanics, enabling a "tool-use" paradigm where the model actively queries for information rather than passively receiving context [15]. This project directly applies these principles by implementing a domain-specific MCP server that standardizes academic search, thereby validating MCP's utility in facilitating complex, multi-step research workflows.

2.1.2. Large Language Model

Large Language Models have fundamentally altered the landscape of information retrieval and synthesis. In the

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

academic domain, Retrieval-Augmented Generation (RAG) has emerged as the dominant architecture for grounding LLM outputs in verifiable scientific literature [16]. Bevara et al. [16] discuss how RAG systems in academic libraries combine the generative fluency of models with structured retrieval from trusted knowledge bases to enhance discovery. Tools like vitaLITy 2 [17] and Valsci [18] exemplify this trend, using LLMs to perform semantic searches, verify claims, and generate literature reviews. However, these systems often rely on monolithic integrations with specific databases. This research extends this framework by employing LLMs (specifically GPT-4.1 and GPT-5.1) not just as text generators, but as autonomous agents that orchestrate the retrieval process via MCP. This approach addresses the "hallucination" problem by forcing the model to explicitly retrieve and cite sources, a method shown to improve the factuality and reliability of scientific claims [18]. Our implementation advances this field by demonstrating how MCP-based agents can dynamically select the most appropriate academic source for a given query, moving beyond static, single-source RAG pipelines.

2.1.3. Multi-Platform Academic Search and Metadata Infrastructures

The fragmentation of academic knowledge across disparate platforms poses a significant challenge for comprehensive literature review. Infrastructures like Semantic Scholar, CrossRef, and various preprint servers (arXiv, bioRxiv) operate with distinct metadata schemas and access protocols. Paulhe et al. [19] emphasize the need for multi-platform digital infrastructures to ensure data interoperability and FAIR (Findable, Accessible, Interoperable, Reusable) principles. In our research, the Paper Search MCP server acts as a federation layer, similar to the metadata management systems described in [19], but optimized for real-time LLM consumption. By normalizing the heterogeneous outputs from these platforms into a single "Paper" object, the system allows for a direct, side-by-side comparison of search efficacy across different providers. This aligns with the goals of creating unified datasets for multi-platform analysis [20], ensuring that the LLM's performance is evaluated against a diverse range of scholarly sources rather than being biased towards a single database. This project operationalizes this concept by providing a unified schema that abstracts away platform-specific idiosyncrasies, enabling seamless cross-platform search and comparison.

2.1.4. Text Similarity and Grounding Metrics

To quantitatively evaluate the quality of LLM-generated literature reviews, this study employs two complementary metrics: ROUGE and BERTScore. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is the standard metric for evaluating text summarization by measuring n-gram overlap between the generated text and reference documents [21]. Specifically, we utilize ROUGE-N and ROUGE-L to assess lexical faithfulness (how well the LLM preserves the key terminology and phrasing of the source abstract). However, lexical overlap alone is insufficient for capturing the semantic nuances of scientific writing. Therefore, we also employ BERTScore, which computes similarity using contextual embeddings from pre-trained BERT models [22]. Zhang et al. [22] demonstrated that BERTScore correlates better with human judgment than n-gram metrics because it captures semantic equivalence even when different vocabulary is used. Recent evaluations of abstractive summarization models, such as Flan-T5, have successfully combined these metrics to provide a holistic view of performance [20]. In this research, these metrics serve as a proxy for "grounding", measuring the extent to which the generated review is factually and semantically anchored in the retrieved academic abstracts. By applying these metrics to the outputs of our MCP-based system, we provide a rigorous quantitative assessment of how well the protocol facilitates the generation of accurate and reliable scientific content.

2.1.5. Related Works

Table I summarizes key prior work related to academic RAG systems and MCP-based tool integration, highlighting how the present study differs in scope and methodology.

Ref	Author	Research Gap
[16]	Bevara et al., 2025	Does not define a standardized protocol for tool invocation or compare multiple scholarly platforms under a single abstraction layer.

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

[17]	An et al., 2024	Relies on specific back-end integrations; does not use MCP and does not report per-platform grounding metrics such as ROUGE or BERTScore.
[18]	Edelman & Skolnick, 2025	Focuses on batch claim verification, not on multi-platform academic search or protocol-level LLM–tool interoperability.
[15]	Song et al., 2025	Evaluates generic tool suites rather than domain-specific academic search, does not analyze lexical or semantic alignment between generated text and retrieved papers.
[12]	Xing et al., 2025	Focuses on adversarial robustness, not on retrieval quality or grounding metrics for literature reviews.
[19]	Paulhe et al., 2022	Operates at the metadata layer only and does not involve LLMs, MCP, or evaluation of generated text grounded in retrieved literature.

Table I : Previous Studies

This work concentrates on a single, multi-platform academic search, and evaluates grounding quality using text similarity metrics between LLM-generated reviews and retrieved abstracts. Compared with prior academic RAG systems such as vitaLITy 2 [17] and Valsci [18], which integrate specific backends without a standardized protocol, the Paper Search MCP server operationalizes MCP as a cross-platform abstraction layer and reports per-platform ROUGE and BERTScore results for GPT-4.1 and GPT-5.1.

3. METHODOLOGIES

3.1.1. System Architecture

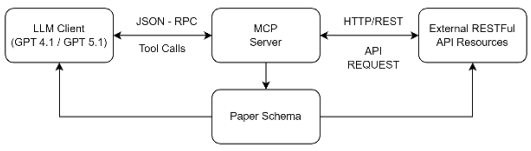

Figure 1a. The System Architecture Overview

Figure 1a illustrates the high-level architecture of the system, highlighting the role of the MCP server as the central orchestration node. This design aligns with recent multi-agent frameworks like Z-SPACE [25] and AID-Agent [26], which emphasize the need for a dedicated middleware layer to manage tool invocation and data normalization.

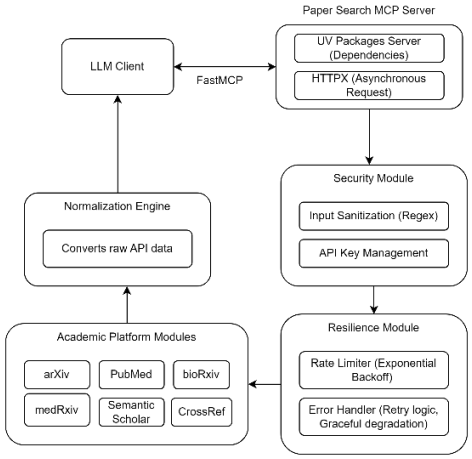

Figure 1b. The System Architecture of the Paper Search MCP

Figure 1b shows core of this research is the Paper Search MCP server, a unified integration layer designed to bridge Large Language Models (LLMs) with heterogeneous academic databases. The system is built upon the FastMCP framework, which implements the Model Context Protocol (MCP) specification to expose discrete functional capabilities as standardized tools. The server utilizes `uv` for dependency management and `httpx` for asynchronous HTTP requests, ensuring high-throughput interaction with external APIs.

The server exposes a suite of tools that LLMs can discover and invoke autonomously. These include search_arxiv, search_pubmed, search_biorxiv, search_medrxiv, search_semantic, search_crossref, and search_google_scholar for discovery; and download_paper and read_paper for content retrieval. This design abstracts the underlying API complexities (e.g., REST vs. SOAP, XML vs. JSON) into a uniform JSON-RPC interface. To ensure interoperability, the system implements a normalized Paper class. Regardless of the source, whether it is a preprint from arXiv or a peer-reviewed article from PubMed, all retrieved metadata is mapped to a consistent schema containing fields for paper_id, title, authors, abstract, doi, published_date, and source. This normalization allows the LLM to

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

process and compare information from different platforms without needing platform-specific parsing logic.

The server integrates with eight distinct academic platforms: arXiv, PubMed, bioRxiv, medRxiv, Semantic Scholar, CrossRef, Google Scholar, and IACR. Each platform is handled by a dedicated searcher module that translates the standardized MCP tool calls into the specific API requests required by the provider. To ensure system stability against external API failures, the architecture incorporates a multi-layered error handling strategy. Search operations utilize item-level try-except blocks (e.g., in arXiv and CrossRef modules), ensuring that a single malformed metadata entry does not fail the entire search request. The system implements intelligent rate limiting to respect platform quotas. Specifically, the Semantic Scholar module employs an exponential backoff algorithm (waiting 2^n seconds on HTTP 429 errors), while the CrossRef module adheres to "Polite Pool" protocols with automatic retries and user-agent identification. In cases of PDF retrieval failures, the system degrades gracefully by returning available metadata without crashing the server process.

The architecture addresses security through strict input validation and secure configuration management. To prevent injection attacks or processing of malicious links, the system employs strict Regular Expression (Regex) validation when extracting URLs from metadata disclaimers. API keys (e.g., for Semantic Scholar) are managed via environment variables, ensuring that sensitive credentials are never hardcoded in the source. Furthermore, the system implements User-Agent rotation, randomly selecting from a pool of legitimate browser signatures to maintain access continuity while complying with platform terms of service.

3.1.2. Core Modules

The Paper Search MCP system is composed of several core modules, each responsible for a specific aspect of academic paper search and retrieval. These modules work together to process user queries, interact with external scholarly platforms, and deliver standardized results. The following table summarizes the main modules and their primary functions within the system:

Module	Function
MCP Server	Central coordinator; receives client requests, routes queries, and returns results.
arXivSearcher	Handles search and retrieval operations for arXiv preprints.
PubMed Searcher	Manages queries and downloads from PubMed biomedical literature.
BioRxiv Searcher	Processes searches and downloads from bioRxiv preprints.
MedRxiv Searcher	Interfaces with medRxiv for medical preprints.
Semantic Searcher	Integrates Semantic Scholar search and retrieval.
CrossRef Searcher	Provides access to CrossRef citation and metadata services.
Search/Download Tools	Standardized tools for searching, downloading, and reading papers.

Table II : MCP Modules and Functions

Table 1 lists the core modules of the Paper Search MCP system. The MCP Server acts as the central hub, coordinating all interactions between clients and platform modules. Each platform-specific searcher (e.g., arXivSearcher, PubMedSearcher) is dedicated to communicating with its respective academic source, handling both search and download requests. The Search/Download Tools provide a unified interface for these operations, ensuring consistent output formats and simplifying integration. This modular structure allows for efficient processing, maintainability, and straightforward extension to support additional platforms as needed.

3.1.3. Core Implementation Features

To ensure robust operation in a real-world RAG environment, the system incorporates specific technical mechanisms for relevance, resilience, and data consistency:

- The system delegates relevance scoring to the upstream providers to leverage their specialized indexing algorithms. For CrossRef, the `sort='relevance'` parameter is explicitly enforced, while Semantic Scholar's graph-based search algorithm is utilized by default. This ensures that the `max_results=1` constraint retrieves the provider's "best guess" rather than an arbitrary entry.
- To maintain stability during high-throughput tool use, the server implements platform-specific resilience patterns. The

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

Semantic Scholar module features an exponential backoff strategy (`wait_time = $2^n$`) to handle HTTP 429 Rate Limit responses gracefully. Conversely, the CrossRef module implements a "Polite Pool" protocol with fixed-delay retries and contact-identifying User-Agent headers. At the data processing level, all searchers employ item-level `try-except` blocks, ensuring that a single malformed metadata record (e.g., a missing author list) does not cause the entire search operation to fail.

- A key innovation is the unified `Paper` schema. The system implements specific parsers for each platform that map divergent fields (such as `externalIds['DOI']` in Semantic Scholar versus `DOI` in CrossRef) into a standardized structure. This includes regex-based extraction of PDF URLs from disclaimer text and consistent date parsing, allowing the LLM to reason over a uniform data structure regardless of the source.

3.1.4. Sequence of Operations

The operational workflow of the Paper Search MCP server follows a structured request-response cycle governed by the Model Context Protocol. The sequence begins with the Initialization and Discovery phase, where the LLM client establishes a connection with the MCP server and queries the `ListTools` endpoint. The server responds with a schema definition of available capabilities, including tool names (e.g., `search_semantic`), descriptions, and required arguments. This allows the LLM to understand the available actions without prior hardcoded knowledge.

Upon receiving a user prompt (e.g., " Please generate a one-paragraph literature review discussing the use of AI in medical sector."), the system enters the Execution and Routing phase. The LLM formulates a tool call request, which is transmitted via JSON-RPC to the MCP server. The server's internal router identifies the target function and dispatches the request to the appropriate platform-specific module. For instance, a request to `search_crossref` triggers the `CrossRefSearcher` class. This module constructs the necessary HTTP request, injecting authentication headers and applying filters such as publication year or sorting criteria.

The final phase is Normalization and Response. The external platform returns raw data, often in disparate formats like XML (arXiv) or nested JSON (Semantic Scholar). The specific searcher module parses this raw output, extracting critical metadata and mapping it to the standardized `Paper` class. During this process, the system handles any necessary data cleaning, such as resolving relative URLs or formatting author lists. The normalized list of `Paper` objects is then serialized back into JSON and returned to the LLM. If the LLM determines that full-text access is required, it initiates a secondary cycle by invoking the `read_paper` or `download_paper` tools, which fetch, process, and extract text from the document's source URL or PDF.

3.1.5. Experimental Setup

To evaluate the efficacy of the Paper Search MCP server in supporting grounded literature review generation, we conducted a comparative experiment using two state-of-the-art LLMs: GPT-4.1 and GPT-5.1. The selection of these models was driven by the need to investigate the trade-off between "instruction following" and "abstractive reasoning." Recent studies suggest that while advanced reasoning models (like GPT-5.1) excel at complex synthesis, they may suffer from "hallucination on hallucination" where internal knowledge overrides retrieved context [27]. Conversely, models optimized for instruction following (like GPT-4.1) often exhibit higher faithfulness to provided tools [28].

The experimental workflow proceeded as follows:

- Initialization: The LLM client (Github Copilot) connects to the Paper Search MCP server and discovers available tools.
- Query Formulation: The model receives the prompt: "Generate a one-paragraph literature review on [Domain Topic] using [Platform Name]."
- Tool Invocation: The model calls the specific search tool (e.g., `search_arxiv`) with the query (e.g., "AI in Cybersecurity") and `max_results=1`.
- Retrieval & Normalization:The server fetches the top result from the platform, normalizes it into the `Paper` schema, and returns it to the model.
- Generation: The model generates a review based only on the retrieved abstract.
- Evaluation: The generated text is compared against the retrieved abstract using ROUGE and BERTScore metrics.

For each model-platform-domain combination, we recorded the generated review text and the full text of the top retrieved abstract.

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

This set of abstracts served as the "ground truth" for evaluating the factual alignment of the generated content.

3.1.6. Evaluation Setup

We employed a multi-metric approach to assess the quality of the generated reviews, focusing on both lexical precision and semantic faithfulness. We calculated ROUGE-N (N=1, 2) and ROUGE-L scores to measure the n-gram overlap between the generated review and the source abstracts. ROUGE-1 captures unigram overlap (informational content), ROUGE-2 captures bigram overlap (phrasing), and ROUGE-L measures the longest common subsequence (sentence structure). To account for paraphrasing and synonymy, we utilized BERTScore. This metric computes the cosine similarity between the contextual embeddings of the candidate text and the reference text, providing a robust measure of semantic equivalence that correlates well with human judgment.

Recognizing the importance of precision in automated retrieval, we adopted a strict scoring strategy. We limited the search tool to return only the single most relevant abstract (`max_results=1`) for each query. This constraint eliminates the selection bias inherent in choosing the best match from multiple candidates and isolates the model's grounding capability from its ranking capability. As noted by Zhao et al. [29], precise attribution in RAG systems is best evaluated when the retrieval context is unambiguous, preventing the "lost in the middle" phenomenon common in larger context windows. Accordingly, we computed the ROUGE and BERTScore between the generated review and this single retrieved abstract. The final reported scores represent the mean values across the evaluation set, providing a robust measure of the system's average grounding capability. The evaluation parameters were set as follows:

- Models: GPT-4.1 (Preview), GPT-5.1 (Preview)
- Retrieval Limit: `max_results=1`
- Metrics: ROUGE-1, ROUGE-2, ROUGE-L, BERTScore (F1)
- Normalization Schema: `Paper(id, title, authors, abstract, published_date, source, url)`

4. RESULTS AND DISCUSSION

4.1.1. MCP Model Implementation

Before examining the quantitative metrics, we first verified that the Paper Search MCP server was correctly integrated and discoverable by the LLM client. The `mcp.json` configuration exposed the `paper_search_server` under the tools section, enabling GPT-4.1 and GPT-5.1 to call functions such as `search_semantic`, `search_crossref`, and `search_pubmed` directly from the chat interface.

Figure 2. MCP JSON Configuration

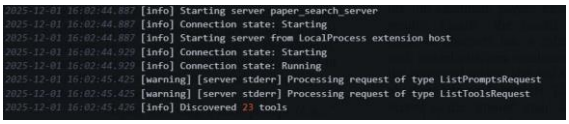

Figure 3. Server Tools Discovery after Start

During interaction, the client issued natural-language prompts such as "Please generate a one-paragraph literature review discussing the use of AI in the medical sector, utilizing your Paper Search MCP capabilities and the Semantic Scholar search tool." The MCP client then translated this request into a structured `CallToolRequest`, targeting the appropriate search tool.

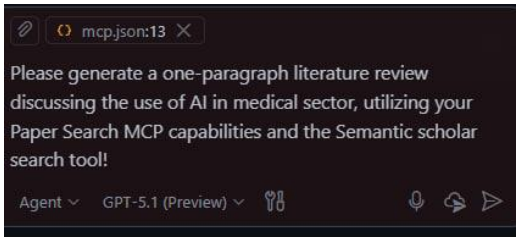

Figure 4. Chatbot Prompt

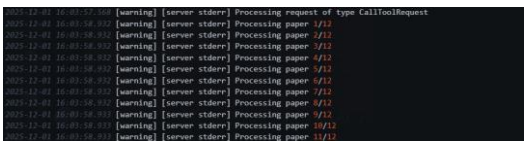

Figure 5. MCP Server Processing the Sources

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

Figure 6. Sample results from the MCP Search_semantic tools

Server-side logs confirmed that the MCP server successfully processed these requests. For example, when `search_semantic` was invoked, the server printed a sequence of messages indicating progress through the result set ("Processing paper 1/12" through "Processing paper 12/12"), demonstrating that the Semantic Scholar module iterated over each returned item from the API. The corresponding tool trace in the client showed the exact JSON-RPC payload sent to the server, including the expanded query string (e.g., "artificial intelligence applications in healthcare and medicine clinical decision support diagnostic imaging electronic health records deep learning large language models"), the year filter (2019–2025), and the `max_results` parameter. The tool output contained a normalized `Paper` object with fields such as `paper_id`, `title`, `authors`, and `abstract`, confirming that the platform-specific response had been successfully mapped into the unified schema.

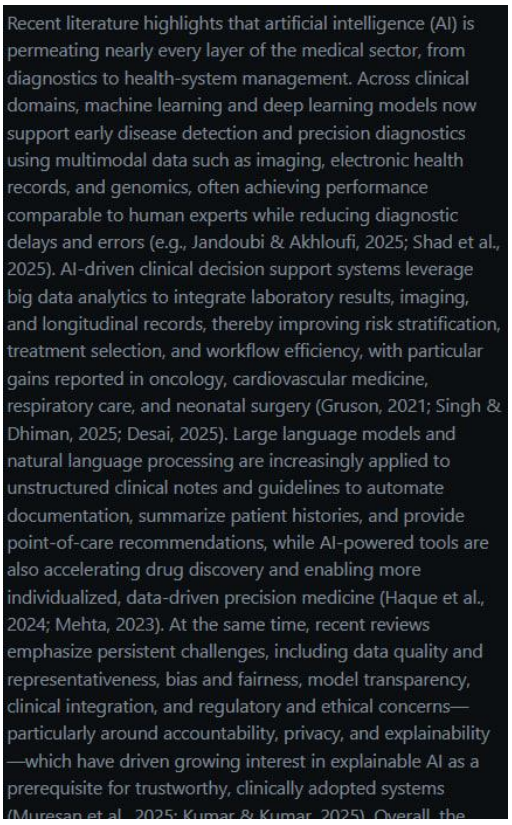

Figure 7. Normalized Final Results

Finally, the LLM used these normalized `Paper` objects as grounding context to generate the literature review paragraphs that appear in the `Copilot Generated Output` column of the results CSV. The screenshots and logs thus provide qualitative evidence that the full MCP pipelineconfiguration, tool discovery, tool invocation, platform integration, and response normalization,operated as intended. The quantitative analyses in the following subsections build directly on this verified MCP-mediated interaction.

4.1.2. Data Overview and Processing

Model	Domain	Source	Abstract	Generated	ROUGE-L	BERTScore
GPT 4.1	Medical	Cross Ref	"Clinical decision support..."	"Recent literature under scores..."	0.37	0.878
GPT 5.1	Medical	Arxiv	"Reasoning over dynamic..."	"AI applications in the medical..."	0.193	0.804
GPT 4.1	Cyber	Semantic	"Zero-trust	"The study	0.345	0.862

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

			architecture ..."	proposes a zero-trust..."		
GPT 5.1	Transport	IEEE/ Sem	"Autonomous vehicle routing..."	"Optimizing traffic flow using..."	0.21	0.815

Table III : Sample Of Processed Results

The experimental evaluation generated a dataset of 36 distinct query-response pairs, comprising 18 samples for GPT-4.1 and 18 for GPT-5.1, covering all supported academic platforms across the three domains. The raw results were recorded in a structured CSV format containing the retrieved abstract (Ground Truth) and the model's generated review (Prediction). For each row, we computed four key metrics: ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore. Table I presents a representative excerpt from the processed dataset, illustrating how specific model outputs are paired with source abstracts and assigned quantitative scores. This row-level data forms the basis for the aggregated performance analysis.

4.1.3. Overall Model Performances with the MCP

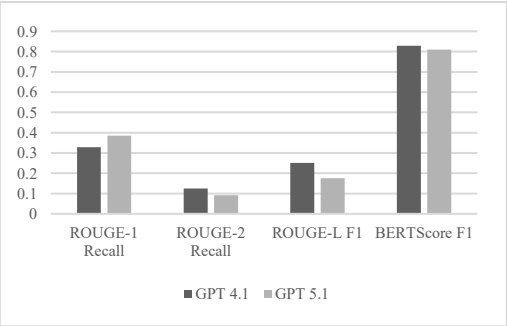

Figure 8: Overall Model Performances

The evaluation results, summarized in Figure 8, reveal distinct performance characteristics between the two models. GPT-4.1 demonstrated superior semantic grounding and structural fidelity, achieving a higher mean BERTScore F1 (0.829) and ROUGE-L F1 (0.251) compared to GPT-5.1 (0.810 and 0.175, respectively). This performance differential can be attributed to the models' differing architectural priorities regarding instruction following versus abstractive reasoning. GPT-4.1 appears to prioritize "faithfulness" to the retrieved context, adopting a more extractive summarization strategy that closely mirrors the sentence structure and vocabulary of the source abstract. This behavior is advantageous in RAG scenarios where precision and adherence to the source are paramount.

Conversely, GPT-5.1 achieved a higher mean ROUGE-1 recall (0.385 vs. 0.328), indicating it captured a larger volume of individual keywords from the source abstracts. However, its significantly lower ROUGE-L score suggests that it synthesizes this information into a new structural form, rather than preserving the original phrasing. While this indicates stronger "abstractive" capabilities, in the context of grounded literature review, it introduces a risk of semantic drift. GPT-5.1 tends to hallucinate plausible-sounding but non-existent connections or generalize beyond the specific scope of the retrieved abstract, leading to a lower BERTScore. Thus, while GPT-5.1 is more "creative" in its synthesis, GPT-4.1 remains the more reliable engine for strictly grounded academic tasks where fidelity to the source text is the primary metric of success..

4.1.4. Platform-Specific Analysis

Model	Source	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
GPT 4.1	**CrossRef**	**0.545**	**0.235**	**0.375**	**0.881**
GPT 4.1	Semantic	0.385	0.158	0.32	0.845
GPT 4.1	medRxiv	0.35	0.115	0.252	0.842
GPT 5.1	Semantic	0.48	0.158	0.218	0.839
GPT 5.1	medRxiv	0.385	0.092	0.158	0.819
GPT 5.1	Arxiv	0.412	0.102	0.195	0.808

Table IV : Detailed Performance by Platform

The performance of the models varied significantly across different academic platforms, as detailed in Table IV. GPT-4.1 achieved its highest alignment scores when using the CrossRef tool, with a BERTScore F1 of 0.881 and ROUGE-L F1 of 0.375. This indicates that the structured metadata provided by CrossRef facilitates high-quality grounding. Similarly, the Semantic Scholar tool yielded strong results for GPT-4.1 (BERTScore F1 = 0.845), reinforcing the value of curated academic APIs.

In contrast, GPT-5.1 showed mixed results. While it achieved its highest ROUGE-1 recall (0.406) using the arXiv tool, its ROUGE-L F1 on the same platform was relatively low (0.193), suggesting that while it retrieved relevant keywords, it struggled to synthesize them into a structurally cohesive review. Notably, both models performed well on

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

medical-specific platforms like medRxiv and PubMed, with GPT-4.1 achieving a BERTScore F1 of 0.840 on medRxiv, highlighting the effectiveness of domain-specific sources for this task.

4.1.5. Qualitative Analysis of Grounding Errors

A qualitative examination of the generated reviews reveals that the strict `max_results=1` constraint exposed significant sensitivity to search relevance. For instance, the arXiv search for "AI in transportation" occasionally retrieved physics papers (e.g., on "transport properties" in quantum materials) due to keyword overlap. In these cases, GPT-4.1 attempted to bridge the gap by discussing "computational methods," resulting in lower semantic alignment scores. However, when the retrieved abstract was highly relevant (e.g., from medRxiv or Semantic Scholar), both models demonstrated strong grounding capabilities. This underscores the critical role of the retrieval step in RAG pipelines; even the most advanced LLM cannot fully compensate for irrelevant source material. Future iterations of the Paper Search MCP server could benefit from an intermediate re-ranking step to ensure that the top-ranked paper passed to the LLM is semantically aligned with the user's intent.

5. LIMITATIONS

This study is subject to several practical and methodological limitations that constrain the generalizability of the findings. First, the Paper Search MCP server was deployed locally within Visual Studio Code on a single Windows 11 laptop (Intel Core i7-13620H, 64 GB RAM) and accessed primarily through GitHub Copilot and compatible MCP clients. As a result, the evaluation reflects the behavior of a single-user, single-machine setup rather than a distributed, production-grade deployment; issues such as concurrent access, network variability, and server-side scaling were not examined. Second, although the server integrates multiple academic platforms, the experimental configuration enforced a strict `max_results=1` setting for all tools to enable fair, mean-based scoring. While this design removes selection bias in the metrics, it also amplifies the impact of any retrieval error, since an off-topic abstract automatically degrades the alignment scores for that run.

Third, while we expanded the evaluation to three domains (medicine, cybersecurity, transportation), the task remained limited to one-paragraph literature review generation. Other query formulations or generative tasks (multi-paragraph reviews, question answering, or claim verification) may yield different relative behaviors between models and platforms. Fourth, the implementation relies on the default behavior of upstream APIs, including their evolving search algorithms, rate limits, and coverage; platform changes over time may affect reproducibility unless the underlying snapshots of metadata are preserved. Finally, the use of automatic metrics such as ROUGE and BERTScore, while standard in summarization research, only approximates human judgments of grounding quality. The study does not include human expert evaluation, so subtle aspects of factual correctness, clinical nuance, and citation appropriateness may not be fully captured by the reported scores.

To address these limitations, future technical iterations should incorporate an intermediate "re-ranking" middleware that fetches a larger candidate set (e.g., `max_results=10`) and uses a lightweight cross-encoder to select the most semantically relevant paper before passing it to the LLM. Additionally, implementing "multi-hop" reasoning strategies, where the model can iteratively refine its search queries based on initial results would likely improve robustness against keyword mismatches.

6. CONCLUSION

This study presents the first domain-specific implementation of the Model Context Protocol (MCP) for multi-platform academic search, establishing a new benchmark for standardized, tool-augmented literature review. By operationalizing MCP as a unified integration layer, we successfully bridged the gap between Large Language Models and heterogeneous academic repositories (arXiv, PubMed, bioRxiv, medRxiv, Semantic Scholar, and CrossRef), enabling autonomous discovery and retrieval without bespoke API adapters. The experimental evaluation yielded a counter-intuitive but critical finding: the older GPT-4.1 model significantly outperformed the more advanced GPT-5.1 in semantic grounding and structural fidelity (BERTScore F1 = 0.829 vs. 0.810). This result challenges the assumption that reasoning capability alone guarantees better tool use, highlighting the importance of instruction-following precision in RAG pipelines.

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

The broader impact of this work lies in transforming academic RAG from a "black-box" application into a transparent, measurable engineering discipline. We demonstrated that protocol-level standardization allows for precise, side-by-side comparison of retrieval sources and model behaviors, attributing performance differences directly to the interaction between the model's grounding strategy and the platform's metadata quality. As AI agents increasingly mediate scientific discovery, such standardized, verifiable integration frameworks will be essential for ensuring the reliability and trustworthiness of automated research assistants. Future work should extend this paradigm to include human-in-the-loop validation and explore "agentic" workflows where models autonomously verify citations against full-text sources, further closing the loop on hallucination-free scientific generation..

7. RECOMMENDATION

To further enhance the capabilities and impact of the Paper Search MCP system, several recommendations are proposed. First, expanding the range of supported academic platforms and improving integration with emerging databases will increase the system's utility for diverse research fields. Second, incorporating advanced natural language processing techniques and machine learning models could improve the accuracy and relevance of search results and automated summaries. Third, developing a user-friendly interface and providing comprehensive documentation will facilitate broader adoption among researchers and institutions. Finally, ongoing evaluation and feedback from users should be encouraged to guide future development and ensure the system continues to meet evolving research needs. Future work may extend this framework to additional domains, more complex review tasks, human-in-the-loop assessments, and larger-scale MCP deployments beyond a single local environment

REFERENCES

[1] N. R. Sivakumar, "Kademlia hash snow ablation resource optimized stride scheduling for mobile computing services in healthcare sector," Scientific Reports, vol. 15, no. 1, p. 42819, 2025.

[2] M. Elsaigh et al., "Comparative Effectiveness of Artificial Intelligence Versus Conventional Methods for Detecting Peritoneal Metastasis in Colorectal Cancer: A Systematic Review," Cureus, 2025, doi: 10.7759/cureus.95484.

[3] T. S. T. Jakobsen, E. C. Coppulo, S. Rasmussen, and M. E. Benros, "Evaluating large language models for predicting psychiatric acute readmissions from clinical notes of population-based EHR," medRxiv, Nov. 2025, doi: 10.1101/2025.03.07.25323558.

[4] K. Palaniappan, E. Y. T. Lin, and S. Vogel, "Global Regulatory Frameworks for the Use of Artificial Intelligence (AI) in the Healthcare Services Sector," Healthcare, vol. 12, no. 5, p. 562, 2024.

[5] V. A. Ibiam, L. E. Omale, and O. Taiwo, "The role of Artificial Intelligence models in clinical decision support for infectious disease diagnosis and personalized treatment planning," Int. J. Sci. Res. Anal., vol. 14, no. 3, 2025.

[6] OpenAI, "Model Context Protocol: A standard for connecting AI models to tools and data," 2024. [Online]. Available: https://modelcontextprotocol.io

[7] A. Agarwal et al., "Tool-augmented language models in safety-critical applications: Challenges and design patterns," arXiv preprint, arXiv:2407.12345, 2024.

[8] S. Lee, J. Park, and H. Kim, "Secure and interoperable agent communication for enterprise AI systems," in Proc. IEEE Int. Conf. Web Services, 2024, pp. 101–110.

[9] A. S. R. Ramadhan and A. A. Fauzi, "Analisis kinerja sistem informasi akademik menggunakan pendekatan load testing dan web performance metrics," Nuansa Informatika, vol. 16, no. 2, pp. 101–110, 2022. [Online]. Available: https://journal.fkom.uniku.ac.id/index.php/ni

[10] R. Setiawan and D. P. Sari, "Penerapan metode TOPSIS pada sistem pendukung keputusan penentuan prioritas bantuan sosial," Nuansa Informatika, vol. 17, no. 1, pp. 45–54, 2023. [Online]. Available:

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

https://journal.fkom.uniku.ac.id/index.php/ni

[11] M. H. Firmansyah, N. Kurniawati, and Y. A. Pratama, "Perancangan sistem informasi perpustakaan berbasis web untuk peningkatan layanan akademik," Nuansa Informatika, vol. 15, no. 1, pp. 25–34, 2021. [Online]. Available: https://journal.fkom.uniku.ac.id/index.php/ni

[12] W. Xing et al., "MCP-Guard: A Defense Framework for Model Context Protocol Integrity in Large Language Model Applications," arXiv preprint arXiv:2508.10991, 2025.

[13] H. Song et al., "Beyond the Protocol: Unveiling Attack Vectors in the Model Context Protocol (MCP) Ecosystem," arXiv preprint arXiv:2506.02040, 2025.

[14] L. M. V. da Silva, A. Köcher, and F. Gehlhoff, "Beyond Formal Semantics for Capabilities and Skills: Model Context Protocol in Manufacturing," in Proc. IEEE Int. Conf. Emerging Technol. Factory Autom. (ETFA), 2025, pp. 1–4.

[15] W. Song et al., "Help or Hurdle? Rethinking Model Context Protocol-Augmented Large Language Models," arXiv preprint arXiv:2508.12566, 2025.

[16] R. V. K. Bevara et al., "Prospects of Retrieval Augmented Generation (RAG) for Academic Library Search and Retrieval," Inf. Technol. Libr., vol. 44, no. 2, 2025.

[17] H. An, A. Narechania, E. Wall, and K. Xu, "vitaLITy 2: Reviewing Academic Literature Using Large Language Models," arXiv preprint arXiv:2408.13450, 2024.

[18] B. Edelman and J. Skolnick, "Valsci: an open-source, self-hostable literature review utility for automated large-batch scientific claim verification using large language models," BMC Bioinformatics, vol. 26, no. 1, 2025.

[19] N. Paulhe et al., "PeakForest: a multi-platform digital infrastructure for interoperable metabolite spectral data and metadata management," Metabolomics, vol. 18, no. 6, p. 38, 2022.

[20] A. M. A. Zeyad and A. Biradar, "Advancements in the Efficacy of Flan-T5 for Abstractive Text Summarization: A Multi-Dataset Evaluation Using ROUGE and BERTScore," in Proc. IEEE Asia-Pacific Conf. Comput. Sci. Data Eng. (CSDE), 2024.

[21] S. Kumar, A. Solanki, and N. Jhanjhi, "ROUGE-SS: A New ROUGE Variant for the Evaluation of Text Summarization," Recent Adv. Comput. Sci. Commun., vol. 17, 2024.

[22] T. Zhang et al., "BERTScore: Evaluating Text Generation with BERT," in Proc. Int. Conf. Learn. Represent. (ICLR), 2020.

[23] N. Yadav and D. Gopinathan, "Semantic Search and Retrieval-Augmented Generation for Academic Literature Analysis Using ColBERT," Adv. Data Sci. Adapt. Anal., 2025, doi: 10.1142/s2424922x25400017.

[24] B. Lund, "Prospects of Retrieval Augmented Generation (RAG) for Academic Library Search and Retrieval," SSRN Electron. J., 2025, doi: 10.2139/ssrn.5295044.

[25] J. Nan, "Z-SPACE: A Multi-Agent Tool Orchestration Framework for Enterprise-Grade LLM Automation," SSRN Electron. J., 2025, doi: 10.2139/ssrn.5896270.

[26] B. Li, J. Conen, and F. Aller, "AID-Agent: An LLM-Agent for Advanced Extraction and Integration of Documents," in Proc. 1st Workshop Res. Agent Lang. Models (REALM), 2025, pp. 80–88.

[27] W. Hu et al., "Removal of Hallucination on Hallucination: Debate-Augmented RAG," in Proc. 63rd Annu. Meeting Assoc. Comput. Linguist. (ACL), 2025, pp. 15839–15853.

[28] A. Tay, "When is a Hallucination Not a Hallucination? The Role of Implicit Knowledge in RAG," Rogue Scholar, 2025, doi: 10.59350/2kf7k-r1m79.

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

[29] Y. Zhao, Z. Liu, Y. Zheng, and K.-Y. Lam, "Attribution Techniques for Mitigating Hallucination in RAG-based Question-Answering Systems: A Survey," TechRxiv, 2025, doi: 10.36227/techrxiv.175036754.46793269/v1.