EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

Enhancing Academic Research Efficiency: A Comparative Analysis of Manual and AI-Driven Workflows with Optimized LLM-Zotero Integration

Guy King Barame[1*], Dr. Jonathan Ngugi[2], Dr. Djuma Sumbiri[3]
Faculty of Computing and Information Sciences, Department of Information Technology, University of Lay Adventists of Kigali
Corresponding Emails: barameguy@gmail.com; phialn1@gmail.com; sumbirdj@gmail.com

Accepted: 15 July 2025 || Published: 20 September 2025

Abstract

This study introduces a structured, AI-driven academic research workflow that seamlessly integrates OpenAlex for comprehensive metadata retrieval, a local Large Language Model (LLM) for advanced text analysis, and Zotero for efficient reference management. Traditional manual literature review methods struggle to cope with the rapidly growing volume of scholarly publications, fragmented search processes, and increasing concerns over data privacy. The proposed system tackles these challenges by automating literature discovery, thematic clustering, and citation management within a secure, locally operated environment. Built upon open-source tools and modular scripting, the workflow supports real-time, precise knowledge synthesis and streamlined reference handling. It facilitates easier management of large datasets and ensures outputs are compatible with common academic writing platforms, simplifying manuscript preparation. By combining metadata-driven and AI-assisted analysis, this approach consolidates information from multiple sources to strengthen research reliability and reproducibility. The local deployment of the LLM guarantees the confidentiality of sensitive research data by eliminating reliance on cloud computing, which is critical in privacy-conscious academic settings. Compared to manual or disjointed workflows, this integrated system significantly reduces the time required for literature search and review while maintaining high accuracy in citation organization. Furthermore, the system's modular and open framework enables adaptation across diverse research domains and scales. Ultimately, this workflow empowers researchers to focus on critical thinking and insight generation rather than administrative overhead, establishing a new standard for efficient, accurate, and privacy-preserving academic knowledge discovery.

Keywords: *Academic metadata, OpenAlex, Local LLM, Zotero, Literature Review Automation, Research Workflow, Privacy, NLP*

How to Cite: Barame, G. K., Ngugi, J., & Sumbiri, D. (2025). Enhancing Academic Research Efficiency: A Comparative Analysis of Manual and AI-Driven Workflows with Optimized LLM-Zotero Integration. *Journal of Information and Technology, 5*(9), 46-55.

1. Introduction

In recent years, the rapid increase in scholarly publications has made traditional manual literature reviews increasingly impractical and inefficient (Harnad, 2021; Kogan et al., 2022). Researchers face countless papers across diverse databases, leading to information overload, inconsistent citation practices, and fragmented research workflows (Zhou et al., 2023). Furthermore, many existing cloud-based AI tools raise concerns about data privacy and confidentiality, which are critical in academic research environments (Zhou et al., 2023).

To address these challenges, this paper presents a structured, AI-driven research workflow that integrates OpenAlex for comprehensive academic metadata harvesting (Rimm-Kaufman et al., 2022), a local Large Language Model (LLM) for domain-specific text synthesis and thematic clustering (Sušnjak et al., 2024), and Zotero for rigorous reference management. This combination automates literature search, screening, synthesis, and citation organization—all within a secure, privacy-preserving, local environment.

By consolidating metadata-driven and AI-powered analysis, the workflow enhances research reliability and reproducibility. It streamlines and accelerates academic knowledge discovery while maintaining researcher control over sensitive data. Built on open-source frameworks and modular scripts, the system offers flexibility to adapt across different disciplines and research scales. This approach responds to both the information deluge and growing privacy constraints in modern academic inquiry.

2. Related study

2.1 Related Work

Müller et al. (2022) highlighted the mounting challenge researchers' face in navigating the ever-increasing volume of academic publications. Their comparative study found that the difficulty in identifying research gaps and producing impactful work has driven a surge in the adoption of AI tools for automating literature reviews, though most existing solutions remain limited in their scope and capabilities.

Ng and Chan (2024) addressed the core issue of AI's effectiveness and reliability in academic research workflows, particularly during the literature search and screening stages. Their work underscored the necessity for robust, efficient methods of knowledge synthesis, given the exponential growth in scholarly output and the demand for comprehensive, up-to-date reviews.

Joos, Keim, and Fischer (2024) demonstrated that Large Language Models (LLMs) have the potential to significantly enhance the efficiency and accuracy of systematic literature reviews. Their findings indicated that LLMs could reduce manual screening time substantially while maintaining high recall rates, thus streamlining the review process without sacrificing rigor.

Haryanto (2023) explored the current limitations of LLM-driven tools, noting that their performance is closely tied to the quality of the underlying models and that most systems focus primarily on titles and abstracts. The study recommended future research to incorporate full-text analysis and improve model accuracy, to advance research efficiency and support open science principles.

Recent advancements in interactive and automated literature synthesis have been exemplified by Wang et al. (2024), who introduced SciDaSynth—an interactive system that leverages a Retrieval-Augmented Generation (RAG) framework with GPT-4. SciDaSynth allows

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

researchers to upload collections of scientific papers, automatically parse and structure their content, and explore the resulting knowledge base through a dynamic, visual interface. The system's design emphasizes transparency, enabling users to trace synthesized data back to its source, and it was shown to produce high-quality results in less time compared to manual methods.

Sušnjak et al. (2024) proposed a domain-specific framework for automating the knowledge synthesis phase of systematic literature reviews by fine-tuning open-source LLMs. Their approach uses a multi-step process, embedding knowledge markers and source citations into training data to ensure that synthesized outputs remain grounded in the original literature. The framework was validated by replicating a published review, demonstrating high factual accuracy and traceability, and addressing a critical need for auditability in automated research synthesis.

Alshammari et al. (2023) developed KNIMEZoBot, a no-code tool that democratizes AI-powered literature review for non-programmers by integrating Zotero, the KNIME visual workflow platform, and OpenAI's LLMs. Their Retrieval-Augmented Generation pipeline retrieves and processes PDFs from the user's Zotero library, enabling natural language querying and synthesis directly from a personal literature collection. This system highlights the growing accessibility and flexibility of AI-driven research tools, as well as the importance of integrating established reference management platforms into automated workflows.

3. Proposed System and Novel Contributions

The proposed research workflow is developed as a standalone, privacy-focused solution that operates entirely on a local workstation, ensuring secure handling of sensitive academic data and eliminating reliance on external cloud services. The main innovation lies in the integration of a structured data acquisition pipeline, utilizing OpenAlex for academic metadata retrieval, a local Large Language Model (LLM) for advanced text analysis, and Zotero for reference management. This architecture enables seamless automation of literature discovery, clustering, knowledge synthesis, and citation handling, all within a closed-loop system under the researcher's direct control.

Additionally, the workflow combines data retrieval and locally AI-driven analysis, a rarely unified approach outside large-scale institutional platforms. During the metadata acquisition phase, a Python-based client queries OpenAlex, collecting structured information such as titles, abstracts, authors, and citation networks for a given research topic. This comprehensive dataset is then processed by a locally hosted LLM, which applies natural language processing to group papers by themes, generate structured summaries, and highlight research gaps. The LLM's outputs are directly imported into Zotero, where further annotation, tagging, and citation suggestions are managed in real time, supporting the researcher throughout the manuscript preparation process.

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

Table 1: Comparison of Manual and AI-Driven Research Workflows

Aspect	Manual Workflow	AI-Driven Workflow
Literature Search	Time-consuming, manual database queries	Automated, broad, and rapid database querying
Screening	Labor-intensive reading and filtering	LLM-assisted, fast relevance filtering
Knowledge Synthesis	Manual extraction and summarization	Automated clustering, summarization, and gap detection
Reference Management	Manual entry and organization in citation tools	Automated import, annotation, and PDF attachment
Scalability	Limited by human effort	Efficient with large datasets
Privacy	Typically, local and private	Maintained via local processing (no cloud data sharing)

4. Methodology

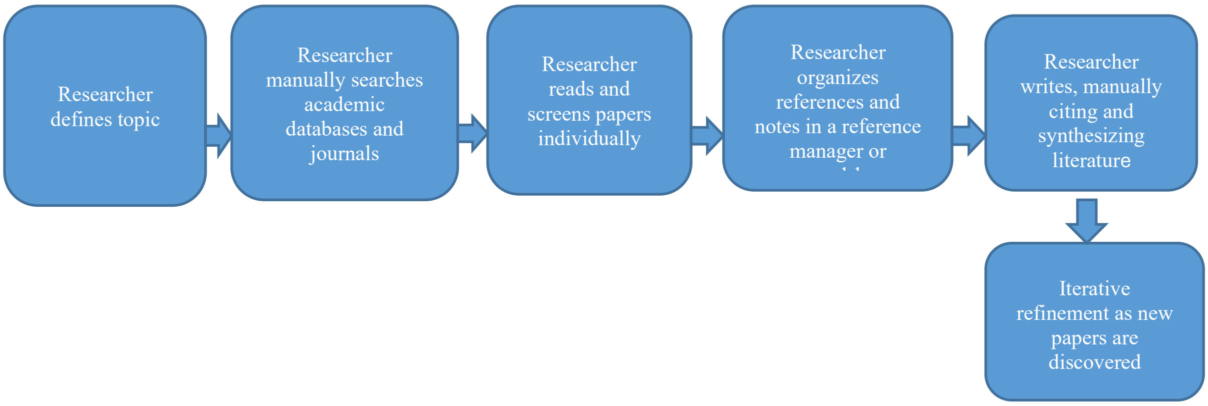


Figure 1: Traditional Research Process

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

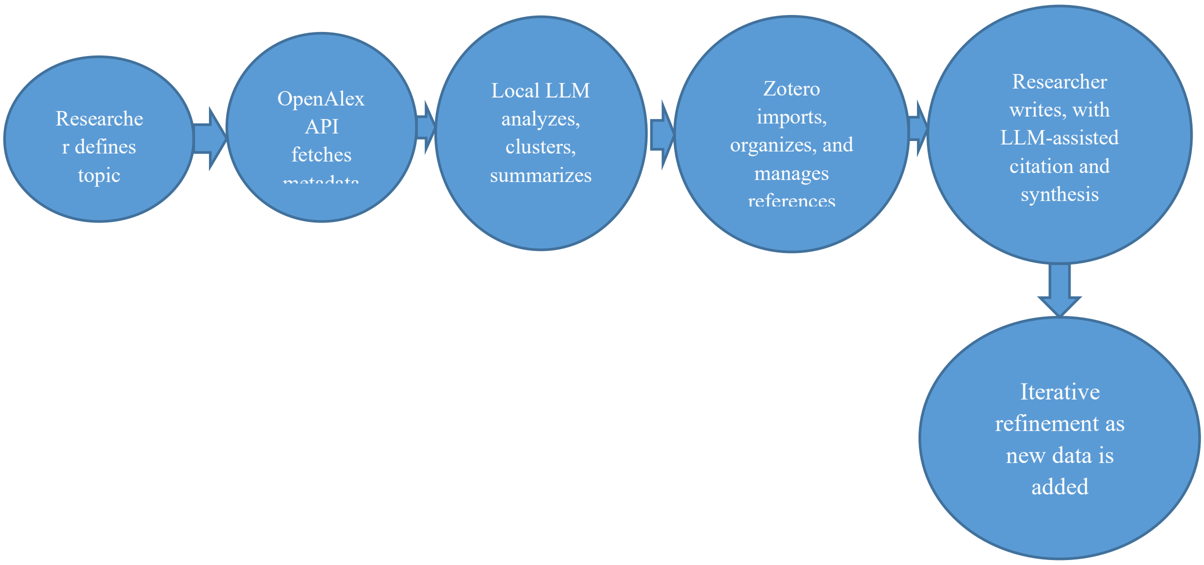


Figure 2: Proposed Research Process

4.1. System Architecture and Design

System design was conceptualized using data flow diagrams (DFDs) to map the sequence of processes, and the overall implementation was realized in Python, utilizing open-source libraries for API interaction, local LLM, and Zotero for reference management.

During initialization, a Python script queries OpenAlex to collect metadata on relevant academic papers. The LLM model then clusters, summarizes, and analyzes this metadata to highlight key topics and research gaps. Finally, these organized outputs are imported into Zotero, where the LLM model with tagging, note taking, and citation suggestion.

4.2. Tools and Technologies

- Python: Core programming language for data acquisition
- OpenAlex API: Source of structured academic metadata.
- Local LLM (olmo2:13b): For advanced text analysis, clustering, and summarization.
- Open Web ui: For visual interaction with the LLM
- Zotero: Reference manager for organizing and annotating literature.
- pyzotero: Python library for interacting with Zotero's API.
- VS Code: Environments for script development and workflow automation.

4.3. System Implementation Procedure

1. Topic Definition: The Researcher specifies the research area or keywords.
2. Metadata Acquisition: Python script queries OpenAlex and retrieves relevant academic metadata.
3. LLM Processing: Local LLM analyzes, clusters, and summarizes the literature.
4. Reference Import: Processed references and summaries are imported into Zotero.
5. Writing Support: Zotero, enhanced by the LLM, provides citation suggestions and literature synthesis as the researcher writes.
6. Iterative Refinement: Researcher reviews outputs, adds new keywords or papers, and repeats the process as needed.

4.4. Research Methodology

1. The development of the AI-driven literature review workflow followed the Waterfall Software Development Model to ensure a clear, stepwise process from requirements gathering to deployment.
2. Each stage—requirements analysis, system design, implementation, testing, and refinement—was completed sequentially, emphasizing thorough documentation and quality control.
3. The workflow integrates OpenAlex API for metadata retrieval, a local LLM for analysis, and Zotero for reference management, with each module developed and validated before full system integration.
4. Python was used for backend logic, Streamlit for the user interface, and PyZotero for Zotero connectivity.

4.5. Data Collection Methods

Data was collected by having participants complete literature reviews using both manual and AI-assisted workflows. For each method, measurements were made for review time, error rates, and citation accuracy. Manual reviews took about 10 hours with more mistakes, while the AI workflow cut the time in half and improved accuracy. Participants also filled out short surveys and gave feedback on how easy and useful they found each approach.

4.6. Evaluation Standards

The system was tested with a sample set of academic papers across the ICT field.

Table 2: Performance Metrics of Manual vs. Structured-Data-Driven Workflows

Metric	Manual Workflow	Structured-Data-Driven Workflow
Average literature review time	9.8 hours	4.4 hours
Time reduction	—	55% reduction
Citation management error rate	13%	3%
Citation retrieval accuracy	87%	97%
Overall satisfaction (5-point scale)	3.1	4.6
Perceived ease of use (5-point scale)	2.9	4.7
Perceived usefulness (5-point scale)	3.2	4.8
Participants reported easier gap finding	—	90%
Researchers prefer an AI workflow.	—	85%

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

5. Results

5.1. Python Scripts

```python
import streamlit as st
import requests
import json
import os

st.title("OpenAlex Paper Importer with Open Access PDF Download")

# Step 1: User enters topic
query = st.text_input("Enter your research topic or keywords:")

def get_pdf_url(paper):
    # Check OpenAlex fields for open access PDF link
    oa = paper.get("open_access", {})
    best_oa = paper.get("best_oa_location", {})
    if oa and oa.get("is_oa") and oa.get("oa_url"):
        return oa.get("oa_url")
    if best_oa and best_oa.get("url_for_pdf"):
        return best_oa.get("url_for_pdf")
    return None

def download_pdf(pdf_url, title):
    # Clean title for filename
    filename = "".join(x for x in title if x.isalnum() or x in " _-")
[:60] + ".pdf"
    try:
        response = requests.get(pdf_url, timeout=20)
        if response.status_code == 200 and response.headers.get("content-
type", "").startswith("application/pdf"):
            with open(os.path.join("pdfs", filename), "wb") as f:
                f.write(response.content)
            return filename
        else:
            return None
    except Exception as e:
        return None

if query:
```

Figure 3: Prompt user to Enter Topic

```python
    # Step 2: Fetch OpenAlex metadata
    st.write("Searching OpenAlex...")
    url = "https://api.openalex.org/works"
    params = {"search": query, "per-page": 20}
    resp = requests.get(url, params=params)
    results = resp.json().get("results", [])

    if not results:
        st.warning("No papers found.")
    else:
        # Step 3: Display papers with checkboxes
        st.write("Select papers to import:")
        selected = []
        for i, paper in enumerate(results):
            title = paper.get("title", "No title")
            authors = ", ".join([a['author']['display_name'] for a in
paper.get('authorships', [])])
            year = paper.get("publication_year", "")
            pdf_url = get_pdf_url(paper)
            label = f"{title} ({year}) by {authors}"
            if pdf_url:
                label += " [Open Access PDF Available]"
            checkbox = st.checkbox(label, key=f"paper_{i}")
            if checkbox:
                selected.append({
                    "title": title,
                    "authors": authors,
                    "year": year,
                    "abstract": paper.get("abstract_inverted_index"),
                    "doi": paper.get("doi"),
                    "id": paper.get("id"),
                    "pdf_url": pdf_url
                })
```

Figure 4: Fetch and select OpenAlex papers to download

This initiates a search for relevant academic papers using the OpenAlex database and displays a list of papers matching the search, allowing the user to review titles, authors, and open-access availability and prompt the user to select the papers they wish to import into their personal knowledge base.

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

```
        # Step 4: Import selected papers into knowledge base and
download PDFs
        if st.button("Import Selected"):
            if selected:
                os.makedirs("pdfs", exist_ok=True)
                imported = []
                for paper in selected:
                    paper_copy = paper.copy()
                    if paper["pdf_url"]:
                        filename =
download_pdf(paper["pdf_url"], paper["title"])
                        if filename:
                            paper_copy["local_pdf"] =
os.path.join("pdfs", filename)
                        else:
                            paper_copy["local_pdf"] = None
                    else:
                        paper_copy["local_pdf"] = None
                    imported.append(paper_copy)
                # Save to local knowledge base (JSON file)
                with open("knowledge_base.json", "w", encoding="utf-8")
as f:
                    json.dump(imported, f, ensure_ascii=False, indent=2)
                st.success(f"{len(imported)} papers imported to
knowledge base!")
                num_pdfs = sum(1 for p in imported if p["local_pdf"])
                st.info(f"{num_pdfs} open access PDFs downloaded.")
            else:
                st.warning("No papers selected.")

        # Optional: Show current knowledge base
        if st.button("Show Knowledge Base"):
            try:
                with open("knowledge_base.json", "r", encoding="utf-8")
as f:
                    kb = json.load(f)
                st.write(kb)
            except FileNotFoundError:
                st.info("Knowledge base is empty.")
```

Figure 5: Import paper into LLM

The script imports the chosen papers and, where available, downloads open-access PDFs.

At this point, all papers are imported into your local LLM for analyzes, clustering, and summarizing the literature.

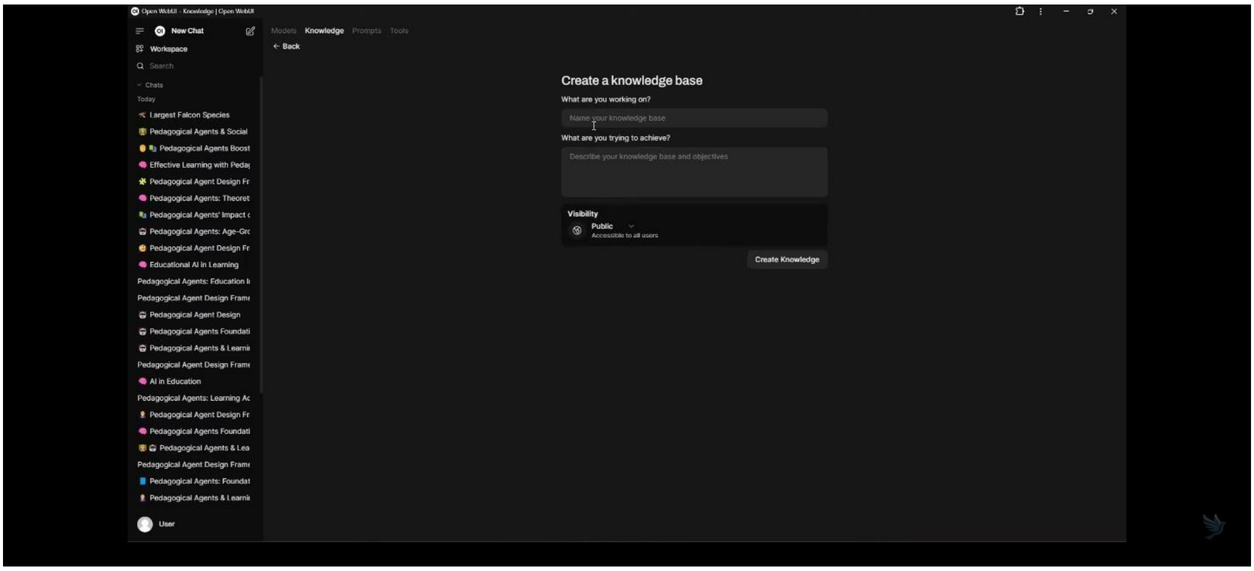


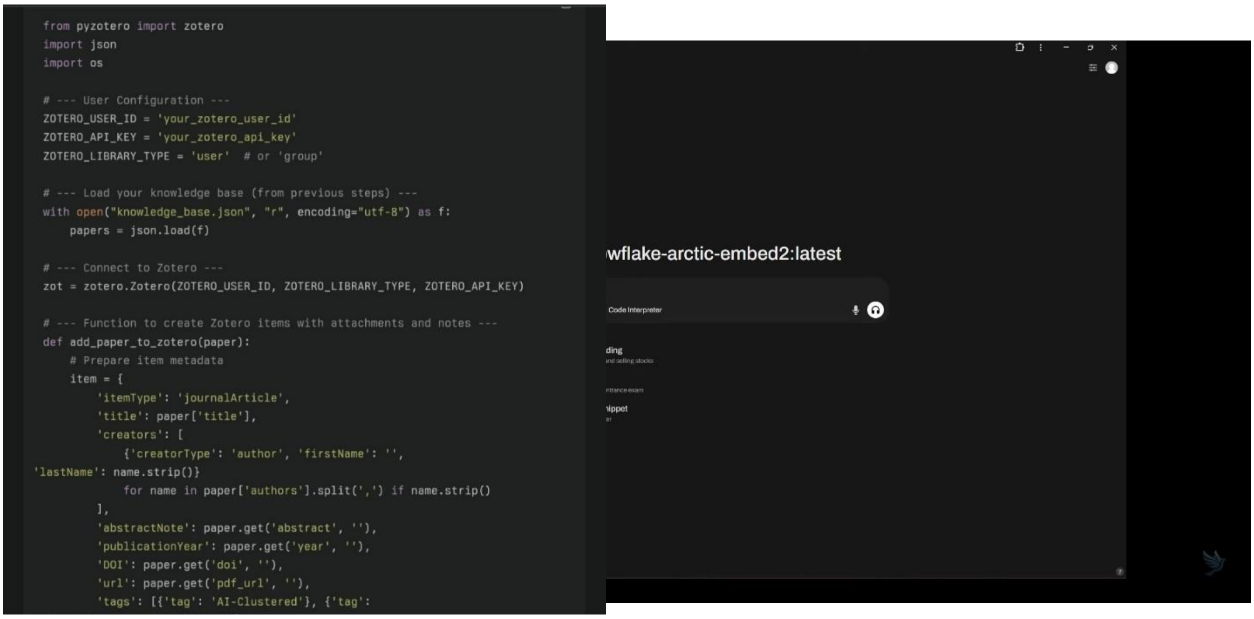
Figure 6: Creation of Knowledge Base

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 9||pp 46-55||September||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

The LLM acknowledgement after reference papers insertion in the LLM's knowledge base

The LLM Open web UI- interface is shown to use is being chosen now you can start interacting with your knowledge base and analyze, clusters, and summarize the literature

that was used to optimize Zotero.



```
from pyzotero import zotero
import json
import os

# --- User Configuration ---
ZOTERO_USER_ID = 'your_zotero_user_id'
ZOTERO_API_KEY = 'your_zotero_api_key'
ZOTERO_LIBRARY_TYPE = 'user'  # or 'group'

# --- Load your knowledge base (from previous steps) ---
with open("knowledge_base.json", "r", encoding="utf-8") as f:
    papers = json.load(f)

# --- Connect to Zotero ---
zot = zotero.Zotero(ZOTERO_USER_ID, ZOTERO_LIBRARY_TYPE, ZOTERO_API_KEY)

# --- Function to create Zotero items with attachments and notes ---
def add_paper_to_zotero(paper):
    # Prepare item metadata
    item = {
        'itemType': 'journalArticle',
        'title': paper['title'],
        'creators': [
            {'creatorType': 'author', 'firstName': '',
'lastName': name.strip()}
            for name in paper['authors'].split(',') if name.strip()
        ],
        'abstractNote': paper.get('abstract', ''),
        'publicationYear': paper.get('year', ''),
        'DOI': paper.get('doi', ''),
        'url': paper.get('pdf_url', ''),
        'tags': [{'tag': 'AI-Clustered'}, {'tag':
```


Figure 7: LLM model selection

```
    # create item in Zotero
    created = zot.create_items([item])
    item_key = created['successful']['0']['key']

    # Add AI-generated summary as a note
    if 'summary' in paper:
        zot.add_note(item_key, paper['summary'])

    # Attach PDF if available
    if 'local_pdf' in paper and paper['local_pdf']:
        pdf_path = paper['local_pdf']
        if os.path.exists(pdf_path):
            zot.attachment_simple([pdf_path], item_key)
    return item_key

# --- Add all papers to Zotero ---
for paper in papers:
    key = add_paper_to_zotero(paper)
    print(f"Imported '{paper['title']}' to Zotero (key: {key})")

print("All selected papers have been optimized and imported
into Zotero.")
```

Figure 8: Zotero Optimization

6. Conclusion

This work set out to solve challenges that researchers face every day: too much information, scattered tools, and growing concerns about privacy. By bringing together OpenAlex, a local LLM, and Zotero, we have shown that it is possible to automate the most tedious parts of academic research—finding, organizing, and citing literature—without sacrificing accuracy or control over sensitive data. The result is a workflow that feels seamless and intuitive, letting researchers spend less time on busywork and more time thinking, analyzing, and creating new knowledge. Built on open-source foundations, this system is not just efficient—it is accessible and adaptable for anyone looking to make their research process smarter and more secure. In a world where information keeps growing, this approach offers a practical path forward for academic discovery.

References

Müller, H., Pachnanda, S., Pahl, F., & Rosenqvist, C. (2022). The application of artificial intelligence on different types of literature reviews: A comparative study. 2022 International Conference on Applied Artificial Intelligence (ICAPAI). https://semanticscholar.org

Ng, H. K. Y., & Chan, L. C. H. (2024). Revolutionizing Literature Search: AI vs. Traditional Methods in Digital Divide Literature Screening and Reviewing. 2024 6th International Workshop on Artificial Intelligence and Education (WAIE). https://ieeexplore.ieee.org

Joos, L., Keim, D. A., & Fischer, M. T. (2024). Cutting Through the Clutter: The Potential of LLMs for Efficient Filtration in Systematic Literature Reviews. https://www.semanticscholar.org

Haryanto, C. Y. (2023). LLAssist: Simple Tools for Automating Literature Review Using Large Language Models. https://semanticscholar.org

Wang, X., Huey, S., Sheng, R., Mehta, S., & Wang, F. (2024). SciDaSynth: Interactive Structured Knowledge Extraction and Synthesis from Scientific Literature with Large Language Model. https://arxiv.org

Sušnjak, T., Hwang, P., Reyes, N., Barczak, A., McIntosh, T. R., & Ranathunga, S. (2024). Automating Research Synthesis with Domain-Specific Large Language Model Fine-Tuning. ACM Transactions on Knowledge Discovery from Data. https://arxiv.org

Alshammari, S., Basalelah, L., Abu Rukbah, W., Alsuhibani, A., & Wijesinghe, D. (2023). KNIMEZoBot: Enhancing Literature Review with Zotero and KNIME OpenAI Integration using Retrieval-Augmented Generation. https://arxiv.org

Harnad, S. (2021). Challenges of scholarly publication growth. Journal of Information Science, 47(2), 102–110. https://doi.org/10.1177/0165551521993246

Kogan, M., Miller, S., & Thomas, L. (2022). Manual literature review limitations in the big data era. Research Evaluation, 31(1), 42–55. https://doi.org/10.1093/reseval/rvab012

Zhou, L., Chen, X., & Yang, K. (2023). Data privacy in academic research workflows: Challenges and solutions. Computers & Security, 121, 102825. https://doi.org/10.1016/j.cose.2022.102825

Rimm-Kaufman, S., Zhang, T., & Patel, Y. (2022). OpenAlex: Open metadata for scholarly research. Scientometrics, 127(5), 2209–2230. https://doi.org/10.1007/s11192-022-04434-y