Advancing FAIR Data Management through AI-Assisted Curation of Morphological Data Matrices

Shreya Jariwala[1, 2], Brooke L. Long-Fox[1, *], Tanya Z. Berardini[1]

[1] Phoenix Bioinformatics, Newark, California, USA

[2] University of California, Berkeley, California, USA

[*] Corresponding Author

Email: blongfox@morphobank.org

Abstract

Curation of biological and paleontological datasets is a labor-intensive process that requires standardization and validation to ensure data integrity. In particular, curation of morphological datasets are prone to human errors such as typographical errors, inconsistent formatting, and incomplete metadata, which hinder reproducibility and compliance with Findability, Accessibility, Interoperability, and Reusability (FAIR) principles. Artificial Intelligence (AI) offers a transformative solution for enhancing research efficiency by automating data validation, improving accuracy, and streamlining curation workflows. This study explores the integration of an AI-based curation tool developed for MorphoBank, an open access repository established to enhance standardization and usability of morphological datasets. We present an AI-assisted curation tool designed to extract, structure, and standardize morphological character data from published literature into the NEXUS file format, a widely used format for phylogenetic analyses. This tool leverages machine learning techniques, including Large Language Models (LLMs), to automate the extraction of character names and states from text in various formats, reducing manual data entry errors and improving data completeness. By integrating this AI-driven approach into MorphoBank, we enhance dataset interoperability and improve metadata accuracy. The AI-assisted curation system significantly accelerates the processing of phylogenetic morphological character datasets while maintaining high accuracy in character extraction and classification. AI is reshaping scientific data management by offering scalable and automated solutions for curating complex phylogenetic datasets. Our findings highlight how AI can assist in ensuring dataset reproducibility, minimizing curation bottlenecks, and improving the accessibility of large-scale morphological datasets. By enhancing compliance with FAIR principles, AI-based curation tools provide researchers with standardized, high-quality data for

evolutionary and taxonomic studies. This study underscores the potential of AI in scientific data curation and its broader implications for future research in paleontology, systematics, and evolutionary biology.

Introduction

Artificial intelligence (AI) is revolutionizing scientific research, including in data curation, where it can enhance efficiency, reproducibility, and accessibility. As biological and paleontological datasets grow in complexity and volume, innovative solutions are needed to standardize, validate, and integrate data effectively. Approaches that use AI offer a promising path forward by automating data extraction, cleaning, and integration processes, reducing human error and ensuring compliance with FAIR (Findable, Accessible, Interoperable, and Reproducible) principles [1]. This study establishes a novel application of AI in scientific data curation, particularly in curating morphological data matrices before uploading them into MorphoBank using information extracted from published journal articles to create files that conform with FAIR and open access standards.

AI in Scientific Workflows

A growing number of scientists are integrating AI tools into their workflows, particularly for tasks such as literature reviews, data analysis, and manuscript writing. Recent surveys indicate that approximately 31% of researchers have used AI chatbots or other AI for research-related tasks [2]. These tools, including large language models (LLMs) and specialized AI-driven software, assist in drafting scientific manuscripts, summarizing literature, and generating code for data processing and visualization [3, 4]. In a study on AI usage among researchers, 22.9%

reported using AI-based tools for research and literature study, and 19.7% for text analysis, processing, and creation [5]. AI's growing role in research highlights its potential to automate routine tasks and improve efficiency across disciplines. While the use of AI is well established in fields such as medical and drug-related research [6, 7], its application in evolutionary biology and morphological data curation is just getting started, including efforts to develop machine-learning pipeline for automated detection of morphometric landmarks [8] and the conceptualization of AI-based digital curators for biological collections [9]. This study presents a novel application of AI for curating morphological phylogenetic character matrices, addressing a critical need for efficient and standardized dataset management in evolutionary research.

Challenges in Morphological Data Matrix Curation

Curation and digitization of biological and paleontological data are essential for enabling large-scale comparative research, yet these processes are often time-consuming and prone to human error [10]. Morphological datasets encode traits (characters) and their observed forms (states) across different taxa, serving as the foundation for many phylogenetic analyses. For example, a character such as fur color in foxes might have states like white, red, or brown. However, the way such characters and states are described varies widely between studies. Some may list states descriptively, others use numeric codes without explanation, and in many cases, state definitions are omitted entirely. Many older morphological character matrices exist only in published literature, often embedded as tables within the paper, requiring researchers to manually transcribe them for reuse. This transcription process is particularly error-prone and introduces challenges in ensuring data accuracy. Manual curation of phylogenetic datasets requires researchers to extract morphological character names and states from published papers and

integrate them into structured formats, such as NEXUS and TNT. The process presents several challenges, including inconsistencies in character descriptions, where variability in terminology and formatting across studies makes data extraction difficult. Additionally, incomplete metadata is a concern, as essential contextual information about characters and states is often missing or inadequately recorded. Furthermore, the risk of transcription errors is high, as manual data entry increases the likelihood of mistakes, compromising dataset integrity and reproducibility. These challenges create bottlenecks in dataset curation and hinder the accessibility and reusability of morphological data. Emerging concepts such as AI-based digital curators offer a promising path forward, emphasizing human and AI collaboration as a way to expand curatorial capacity and improve resilience in data management systems [9].

AI as a Solution for Data Curation

AI provides a scalable and automated solution to these challenges. By applying AI to the curation of morphological datasets, we demonstrate how automated tools can improve data quality by enhancing accuracy and completeness, increase efficiency by reducing the time required for data extraction and standardization, and enhance accessibility by making phylogenetic data more usable for researchers in systematics, paleontology, and evolutionary biology. Morphological data, detailing the physical characteristics of fossils as well as extant organisms is exceptionally valuable as it forms the foundation for understanding evolutionary relationships and organismal interactions through geologic time. Machine Learning (ML) models can extract structured morphological data from scientific literature, standardize formats, and enrich datasets with missing information. AI-based curation tools can also improve interoperability between databases, facilitating seamless data integration across platforms. In this

study, we present an AI-based method for curating phylogenetic matrices, focusing on how an AI model extracts morphological character data from published sources and integrates it into a structured NEXUS file for phylogenetic analyses. This approach is implemented for use in MorphoBank, a web-based repository for morphological data, where it facilitates real-time collaboration and dataset archiving. The developed tool illustrates AI's potential to support large-scale data curation (a normally time-intensive and error-prone process) in the biological sciences and offers a framework for extending AI-driven curation to other disciplines.

NEXUS File Format

The NEXUS file format is a flexible standard used in systematics and phylogenetics to encode different types of data for evolutionary analyses [11]. A typical NEXUS file is divided into blocks, such as TAXA, CHARACTERS, MATRIX, and TREES. The TAXA block lists the taxa, while the CHARACTERS block defines the traits being compared. However, in many published morphological datasets, character definitions are often missing, making interpretation and reuse difficult. The MATRIX block is critical, as it contains the actual scoring of character states recording the value of each character for every taxon in the dataset. While not all blocks are required to run an analysis, the CHARACTERS and MATRIX blocks are essential for morphological data, as they encode both the structure and content of the dataset. The CHARSTATELABELS command is an optional but useful component within the CHARACTERS block, as it allows you to name each character (and optionally its states), making your data matrix more interpretable, especially in morphological datasets where characters represent anatomical features rather than sequence positions. The command lists each

character by number, followed by a label describing the character, and, if desired, a list of possible states for that character.

MorphoBank and AI-Driven Curation

MorphoBank (morphobank.org) is a web application and database permitting real-time collaboration on, and archiving of, morphological matrices for phylogenetics, many of which include primary data from fossil species. Administered by Phoenix Bioinformatics, a non-profit organization specializing in digital research resource sustainability, MorphoBank's goal is to intersect with the efforts of other digital initiatives in the same scientific domain. MorphoBank has partnered with the Paleobiology Database (PBDB) to transfer 595 Nexus files describing phylogenetic research on fossils from that repository to MorphoBank [12]. The matrices are in a NEXUS file format (see above), and each NEXUS file is linked to a peer-reviewed published paper. However, the files lack CHARACTERS blocks, information needed for the data to be fully FAIR. Researchers downloading the file in the original state would not know the names of the characters or the states and would need to backtrack to the original or even older publications to find them. Initial data transfer efforts relied on manual data curation which was a labor-intensive task due to the highly variable structure, number, and length of character names and states. On average, a 100 character matrix could take upwards of two hours to curate manually. To increase efficiency in adding this information to the NEXUS files, an AI-based solution was developed to automate the process of data curation for MorphoBank. The web-based tool streamlines the extraction of morphological character name and state data from published articles and merges it with the matrix-only NEXUS file to create a final file with complete character information.

Materials and Methods

This study developed an AI-based software package, herein named MatrixCurator (https://morphogen.streamlit.app/), that leverages an Large Language Model (LLM) to transform unstructured morphological data from research papers into structured data suitable for phylogenetic analysis. The development process was broken down into the following stages: 1) Data Acquisition and Preprocessing; 2) LLM Selection and Prompt Engineering; 3) Structured Data Schema Definition; 4) Performance Evaluation (Fig. 1).

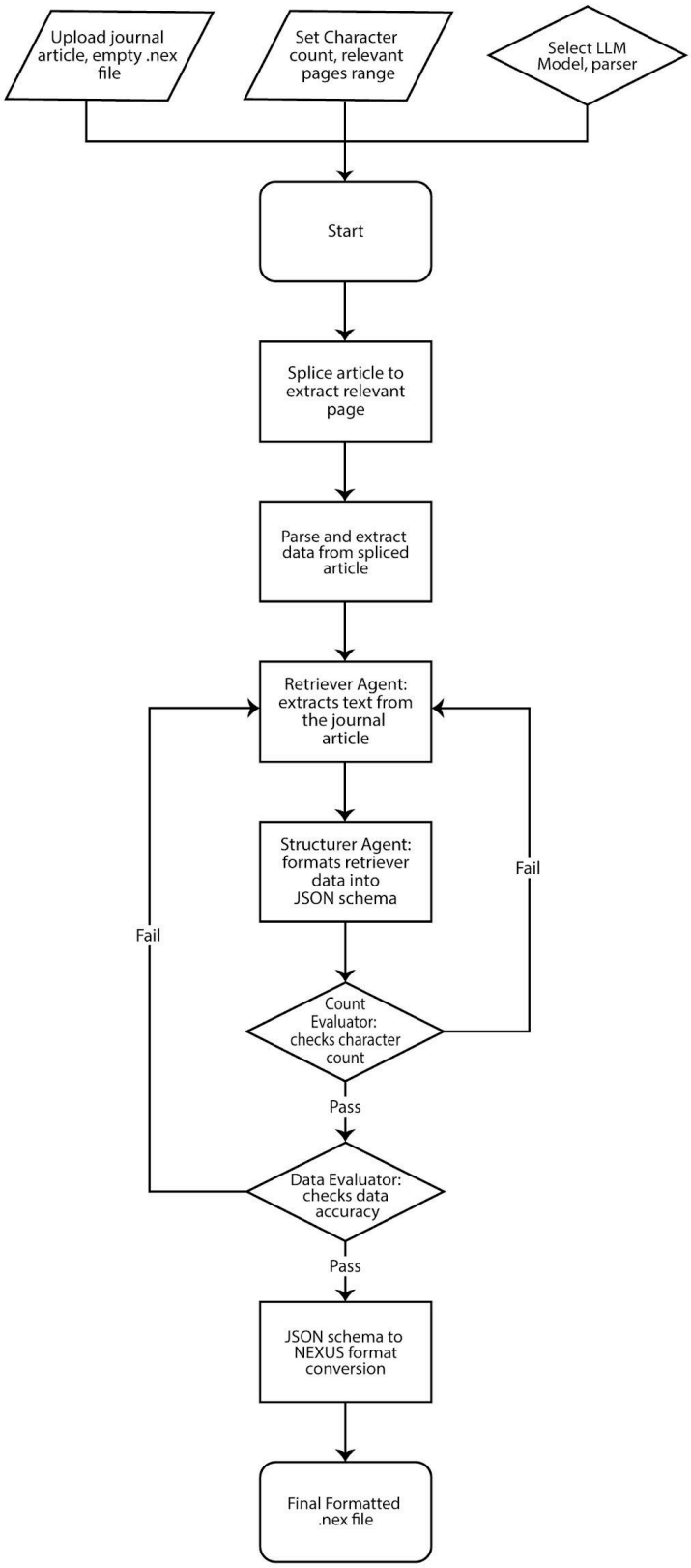
Upload journal article, empty .nex file
Set Character count, relevant pages range
Select LLM Model, parser
Start
Splice article to extract relevant page
Parse and extract data from spliced article
Retriever Agent: extracts text from the journal article
Structurer Agent: formats retriever data into JSON schema
Count Evaluator: checks character count
Fail
Pass
Data Evaluator: checks data accuracy
Fail
Pass
JSON schema to NEXUS format conversion
Final Formatted .nex file

Figure 1: This Large Language Model (LLM)-assisted pipeline automates the conversion of journal articles into the NEXUS (.nex) format through a multi-step process. The workflow begins with user inputs, including the journal article, an empty .nex file, character count specifications, the relevant page range, and the selection of an appropriate and LLM document parser. The relevant pages of the article are then spliced, and parsed for data extraction. A Retriever Agent extracts the characters data, which is subsequently structured into a JSON schema by a Structurer Agent. This structured data undergoes two validation steps: a Count Evaluator checks if the extracted content adheres to character count specifications, and a Data Evaluator verifies the accuracy of the extracted data. If either validation step fails, the process iterates back to the Retriever Agent for re-extraction and re-structuring. Upon successful validation by both evaluators, the JSON schema is converted into the final, formatted NEXUS (.nex) file.

Data Acquisition and Preprocessing

This study uses an AI-curation tool to extract morphological character data from scientific publications and add this data to a NEXUS file for curation in a specialized repository. The source data is typically available as tables within the published manuscript or as supplementary information. The character information is often in PDF or DOCX formats, and these documents often present significant challenges for automated extraction due to inconsistent formatting, where character descriptions and state might be spread across multiple columns and pages. Older documents may consist of scanned images packaged as a PDF. To address these challenges, a robust approach to automated data acquisition and preprocessing was developed.

The preprocessing pipeline begins with a manual inspection of the source document to identify and isolate the page ranges containing relevant morphological character data within each document. Concurrently, the character count within the paper is also manually determined, often starting with an index of 0. Subsequently, each document undergoes parallel processing using three different parsing methods to maximize data retrieval, as no single method is universally optimal given the different ways researchers present data.

The initial step in the preprocessing pipeline involves splitting the document into individual pages containing the identified relevant data using Python 3.12 [13]. The page ranges are then extracted and saved as a new PDF or DOCX corresponding to the original input file type, operations which are done using the PyPDF2 [14] and python-docx [15], respectively. These extracted pages are then independently processed by dedicated parsers. For PDF files, PyMuPDF [14] is utilized to parse the pages in their natural reading order. This parser incorporates Optical Character Recognition (OCR) as a fallback mechanism to handle image-based data, ensuring that information contained in the scanned documents is also extracted. DOCX files are processed using Pandoc [16] to convert them into Markdown format, facilitating downstream parsing. In addition to these community-standard parsing techniques, a generative AI document parser known as LlamaParse [17] is utilized. This approach leverages a Large Language Model (LLM) accessible via the LlamaCloud [17] Application Programming Interface (API) to extract information from documents. Finally, we utilize Gemini's [18] Native Vision features, allowing the LLM to leverage its multimodal capabilities to understand and integrate both text and visual elements to process documents. Gemini's Native Vision capabilities accurately transcribe tables, interpret complex multi-column layouts, understand charts, sketches, diagrams, and handwritten

text within documents, and then use this textual and visual information to carry out end-to-end tasks. DOCX files are first converted to PDF before being processed by Gemini's Native Vision.

Large Language Models (LLMs)

This study leverages the Gemini family of LLMs from Google AI, specifically *gemini-1.5-flash, gemini-2.0-flash, gemini-1.5-pro,* and *gemini-2.0-pro*. Gemini models were chosen for their availability on a pay-per-usage basis and their ability to process large input prompts, offering context windows of 1-2 million tokens depending on the specific model. Tokens are the units of text that LLMs process that are roughly equivalent to words or word parts. A 1-2 million token window allows the model to handle the full text of even very long documents. One million tokens, according to Google, is equivalent to 1,500 pages of text or 30,000 lines of code. This capability is essential for efficiently processing the complete text of large research papers, especially those containing hundreds of characters or detailed studies that span numerous pages. The system is composed of three distinct agents, each with their own specified roles: the Retriever, designed to extract information about each character; the Structurer, which organizes the extracted information as JSON; and the Evaluator, responsible for assessing the quality and accuracy of the extracted data. Due to the faster inference speeds and lower cost, flash models were selected for use in the Retriever and Structurer agents. The high-speed requirements of these agents, which involve numerous inferences for each character within a paper, necessitate the efficiency provided by the flash models. Conversely, the Evaluator agent, requiring more nuanced understanding and reasoning to accurately assess data quality, utilizes the more powerful pro models. The pro model refers to a larger, more robust version of the Gemini LLM, fine-tuned for improved accuracy, reasoning, and handling of longer or more nuanced prompts.

Our methodology employs a multi-agent (i.e., a system involving multiple AI entities working collaboratively to complete complex tasks) approach, meaning we use multiple AI models which are assigned different roles and comprising the three distinct agents: the Retriever, the Evaluator, and the Structurer. Each agent operates based on a combination of a System Prompt, which guides the model's overall behavior, and an Instruction Prompt, which provides specific task instructions, along with data from the research paper being analyzed. To optimize the performance of these prompts, we utilized Vertex AI Prompt Optimizer to automatically refine the system instructions associated with our instruction prompts. The Retriever agent processes the parsed data from the research paper and extracts relevant information specific to each character of interest. The extracted character data is then passed to the Evaluator agent, which assesses the accuracy of the retrieved data by comparing it against the original paper. The Evaluator assigns a score between 1 (lowest accuracy) and 10 (highest accuracy), reflecting the accuracy of the retrieval. This evaluation is performed using LangChain's Scoring Evaluator, which is configured with the correctness criteria. Specifically, the Evaluator receives the original prompt input, the LLM's initial response prediction, and the parsed paper content reference. The evaluation process is implemented as follows. The evaluation process employs a correctness-based scoring function using a preconfigured tool called "labeled_criteria", which is loaded with a specified large language model (LLM). This Evaluator is designed to assess how accurately the model's output aligns with a reference standard. The evaluation is conducted by comparing three components: the original input prompt provided to the model, the model's generated prediction, and a human-verified reference response. The Evaluator analyzes these elements and assigns a numerical score reflecting the degree of correctness of the model's

output. This score is then extracted from the evaluation results and returned for further analysis. Finally, the Structurer agent converts the retrieved character data into a standardized JSON schema, facilitating subsequent analysis and integration.

To enhance efficiency and reduce costs, we implemented the Gemini API's context caching feature. This allows the parsed research paper content to be passed to the model only once, then cache the input tokens, and then refer to these cached tokens for all subsequent requests related to that data source paper. By avoiding repeated transmission of the full paper content for each character to be read, the number of tokens per character request is significantly reduced and, consequently, the cost of each morphological character read request is lowered.

Structured Data Schema

To facilitate computational analysis, we structured the data extracted from the LLM output according to a predefined JSON schema. This schema was designed to represent the states associated with different characters or features. The primary structure consisted of a dictionary where each key, denoted as “Character 1” or “Character 2”, for example, represented a specific character or feature being analyzed. The value associated with each key was an array of strings, with each string representing a distinct ‘state’ or characteristic exhibited by that character. Each character in the JSON schema can have a different number of possible states. For instance, Character 1 may have three states (e.g., "State 0," "State 1," and "State 2"), while Character 2 may have only two (e.g., "State 0" and "State 1"). Subsequently, this JSON structure was cast into the NEXUS format (.nex), a standard format in phylogenetics and evolutionary biology used for representing character data. This conversion was necessary to enable the use of phylogenetic software packages for further analysis. Each character and its associated states were represented using the ‘Enumeration’ keyword, followed by the character name and a forward slash (/)

separating the character from the list of its states. The states were then listed as individual strings, delimited by spaces. This structure provides a clear and standardized way to encode character-state relationships. For example, the JSON data {"Character 1": ["State 0", "State 1", "State 2"]} would be represented in the NEXUS format as 'Enumeration' 'Character 1' / 'State 0' 'State 1' 'State 2'. Similarly, the JSON {" Character 2": ["State 0", "State 1", "State 2"]} would be formatted as 'Enumeration' 'Character 2' / 'State 0' 'State 1' 'State 2'. This process ensured that the information extracted from the LLM was appropriately formatted for subsequent phylogenetic analyses, leveraging the broad compatibility and well-defined structure of the NEXUS format.

Algorithm Optimization

To enhance the accuracy of the character retrieval system, a two-stage evaluation process was implemented. This process consisted of the Evaluator Agent, designed to assess the correctness of retrieved character data, and a Count Evaluator, responsible for verifying the completeness of the character set. The Evaluator Agent directly compared the LLM-retrieved character data against the original data. When the LLM provided an incorrect response, that response was stored, and the LLM was subsequently re-prompted with a negative prompt designed to prevent future iterations of the same error. These negative prompts included explicit corrections paired with firm instructions (e.g., “Do not include anatomical regions not mentioned in the original paper” or “Avoid summarizing character states—list them exactly as shown”). This process aimed to refine the LLM’s retrieval capabilities through iterative learning. Initially, the “flash” model was employed as the Retriever Agent due to its advantage in inference speed and cost-effectiveness. However, after this initial phase, the Retriever Agent was switched to a “pro” model to leverage its enhanced capabilities in handling more complex retrieval tasks.

Complementing the Evaluator Agent was the Count Evaluator. The Count Evaluator performed a simple but crucial function: it counted the total number of characters retrieved by the LLM and compared this count to the user-specified character count. The decision to utilize “flash” models, where applicable, stemmed from their superior inference speed and lower cost, allowing for more efficient evaluation cycles. Conversely, “pro” models were strategically deployed for tasks requiring higher complexity, such as the refined retrieval phase within the Evaluator Agent.

Community Cloud Deployment

The application was deployed to Streamlit Community Cloud. The process involved navigating to the Streamlit user dashboard and clicking "New app," then pasting the link to the application codebase (https://github.com/MorphoEx/morphoex). The correct branch was selected, and streamlit_app.py was designated as the main file. Necessary application secrets, such as API keys, were input via the “Advanced Section.” Finally, initiating the "Deploy" button triggered the build and launch process on Streamlit Cloud, which generated a unique application URL. At present, the deployment is only accessible to internal users and is not publicly available; however, the following section outlines how others can deploy the tool locally using their own API keys.

Linux Host Deployment

To run the application locally, the repository was cloned from GitHub, and the required Python dependencies were installed using pip install -r requirements.txt. The necessary API keys were added to the .streamlit/secrets.toml file. The application was then launched using streamlit run streamlit_app.py. To enable persistent background operation, the command nohup streamlit run

streamlit_app.py & was used, allowing the application to run independently of the terminal session.

Results

Data Extraction Performance

To evaluate document parsing capabilities, the performance of PyMuPDF, Pandoc, LlamaParse, and Gemini Native Vision was benchmarked. This benchmark utilized a curated dataset of 12 research publications, specifically selected because they contain morphological matrices and represent a range of formatting complexities and layout styles. The benchmark compares a base markdown file (**`base_md`**) with the text extracted from a PDF/DOCX file (**`result`**) and calculates a similarity score between the two texts based on inserted, deleted, or replaced to make the two sequences identical. The closer the score is to 1.00, the more similar the texts are. Table 1 presents the average similarity scores for each parsing module. Gemini Native Vision achieved the highest average similarity ratio (0.86), followed by PyMuPDF (0.66), Pandoc (0.59), and LlamaParse (0.46).

Table 1: Average similarity ratio between extracted text and a base markdown file for each parsing module across 12 research publications.

Parser	Similarity Ratio
LlamaParse	0.46
PyMuPDF	0.66

Gemini Native Vision	0.86
Pandoc	0.59

Agent Performance and Efficiency

We measure the performance and efficiency of our multi-agent system by measuring inference time and cost per character. Table 2 summarizes the average inference time and cost for each agent using different Gemini models, both with and without context caching.

Table 2: Average inference time and cost for each agent using different Gemini models, both with and without context caching

Agent	Model	Average Inference Time (s)	Cost per Character (USD, $)
Retriever	gemini-2.0-flash	6.4	0.0001 - 0.0002
Retriever	gemini-1.5-flash	4.8	0.0001 - 0.0002
Retriever	gemini-1.5-pro	6.6	0.0012 - 0.0034
Evaluator	gemini-1.5-pro	6.4	0.0012 - 0.0034
Structurer	gemini-2.0-flash	1.0	0.0001
Structurer	gemini-1.5-pro	2.1	0.0012

As expected, the flash models (gemini-1.5-flash and gemini-2.0-flash) demonstrated superior speed and cost-effectiveness for the Retriever and Structurer agents. The gemini-1.5-flash model achieved the fastest inference time for the Retriever agent (4.8 seconds), slightly outperforming gemini-2.0-flash (6.4 seconds) while maintaining the same low cost per character ($0.0001). This difference in inference time, while relatively small, can become substantial when processing large datasets containing thousands of characters. The Structurer agent, also utilizing gemini-2.0-flash, exhibited a comparable low cost and a fast inference time of 4.5 seconds. Although the gemini-1.5-pro model exhibited a slightly longer inference time (6.6 seconds) for the Retriever agent compared to gemini-2.0-flash, its increased computational power proved beneficial for the Evaluator agent. The Evaluator agent, tasked with the more complex task of assessing data accuracy, leveraged the enhanced reasoning of the pro model. While the inference time for the Evaluator (using gemini-1.5-pro) was comparable to the Retriever using gemini-2.0-flash (6.4 seconds versus 6.4 seconds), the cost per character was significantly higher ($0.0012 versus $0.0001). This trade-off between cost and performance highlights the importance of strategically selecting the appropriate model for each agent based on the specific task requirements.

Computational Efficiency

Leveraging the Gemini API's context caching mechanism significantly reduced the computational cost of character extraction. To illustrate, consider the analysis of a paper with a matrix containing 164 characters. Without caching, each character request required an average of 1,919 tokens, resulting in a total of 314,716 tokens for all 164 characters. With context caching, an initial 1,820 tokens were used to cache the paper's content. Subsequent character requests then

averaged only 119 tokens each, totaling 19,516 tokens for all characters. This resulted in a combined total of 21,336 tokens with caching, a reduction of approximately 93% compared to the 314,716 tokens required without caching. Assuming a token cost of $0.0000015 per token (based on standard API pricing), the cost without caching would be approximately $0.47, while the cost with caching would be only about $0.03. This dramatic decrease in token usage highlights the practical benefits and cost-effectiveness of employing context caching, especially for large-scale data extraction tasks involving numerous character requests within the same document.

Curation of Morphological Data

To date, we have applied our AI-based curation tool to over 300 published papers containing morphological character matrices. Each publication included a minimum of 20 morphological characters, with some datasets exceeding several hundred characters. In total, the tool has processed over 35,000 character-state entries across a wide taxonomic range. The AI system successfully extracted and structured morphological character data from diverse formatting styles, table structures, and text descriptions. Despite significant variation in how characters and states were presented, ranging from fully labeled matrices to minimally described tables, the tool demonstrated high robustness in identifying character names, corresponding states, and associated taxa. Quality checks by expert curators showed that over 90% of extracted character data required only minor or no edits, substantially reducing the time and labor typically required for manual data transcription. Compared to manual transcription, which may take up to 4 hours per dataset, the AI tool reduced processing time by over 90%, significantly accelerating data

reuse workflows. This performance demonstrates the tool's capacity to fill a critical gap in morphological data reuse and to support large-scale, reproducible phylogenetic research.

Discussion

This paper presented an innovative application AI-based method for data mining, parsing, and curation holds great potential for capturing and organizing morphological data, as well as integrating other published datasets. The integration of AI in the curation of morphological phylogenetic datasets offers a transformative approach in data management by enhancing data quality, improving accuracy and completeness, and streamlining the extraction and standardization processes. By utilizing AI-powered tools, we can significantly reduce the time required for data curation, making phylogenetic data more accessible and usable for researchers in systematics, paleontology, and evolutionary biology.

The results of this study demonstrate the potential of AI to fill a critical gap in the curation and reuse of morphological data. However, beyond automation, this work also aligns with a growing vision for AI agents as digital curators that are designed to augment human expertise by engaging in adaptive, task-specific collaboration [19]. These digital curators could take many forms, including conversational assistants powered by large language models (LLMs) trained on collection-specific knowledge. Such tools would enable richer human and AI dialogue, drawing on prior interaction history and adjusting dynamically to the context of current curatorial goals [20]. Rather than functioning as standalone tools, digital curators are envisioned as collaborative partners capable of performing a wide range of tasks from information extraction and metadata enrichment to anomaly detection and hypothesis generation [21,22]. For example, they could

extract trait data from specimen images, decipher phenotypes from multimedia sources, or flag inconsistencies in legacy datasets for human review [23–26]. The selection of appropriate digital curators for specific tasks could itself be AI-assisted, allowing for a flexible and scalable ecosystem of specialized tools [27]. By connecting disparate datasets and surfacing overlooked patterns, digital curators could also support the development of extended digital specimen networks [28] through integrated digital infrastructures that provide a more holistic view of biodiversity. This vision aligns closely with the goals of initiatives like MorphoBank, where standardization, accessibility, and reusability are foundational. The system presented here (an AI-assisted tool for morphological character matrix curation) represents a concrete step toward realizing this broader vision. It demonstrates that with the right infrastructure and human-AI interaction design, curatorial workflows can evolve to become faster, more accurate, and more adaptable to the demands of modern biodiversity research.

AI-assisted curation addresses major challenges in scientific data management including the standardization, reproducibility, and reusability of morphological datasets. Morphological matrices often originate from diverse sources with inconsistent formats and incomplete metadata, making integration and reuse difficult. Our AI-enhanced NEXUS files improve compliance with FAIR principles by enriching metadata, standardizing character and taxonomic information, and producing well-structured, machine-readable datasets [29]. This enables greater interoperability across databases and facilitates comparative research across disciplines. Traditionally, manually curating a data matrix with 120 characters takes approximately two hours; our tool generates the same file in under a minute, with quality control completed in about 20 minutes. This efficiency not only reduces labor but also minimizes human error, improving overall data integrity. For

example, integrating a newly discovered species into an existing matrix is streamlined, with consistent formatting and character-state encoding ensured automatically. These improvements are especially valuable for large-scale datasets. For instance, MorphoBank Project 4446 (http://dx.doi.org/10.7934/P4446) includes 2,633 characters and would be prohibitively time-consuming to duplicate or expand manually. AI tools now allow researchers to adapt such datasets quickly for new studies, increasing their long-term usability. As a repository, MorphoBank plays a critical role in maintaining accessible, replicable morphological data [30], and AI-assisted curation significantly enhances its capacity to support large, heterogeneous datasets. By promoting more efficient workflows and enabling seamless cross-study synthesis, AI integration in MorphoBank marks a meaningful advancement for evolutionary and taxonomic research.

The large scale and complexity of scientific literature posed significant technical challenges for this project. Many papers span 20 to 50 pages and contain extensive tables, figures, and unstructured text, making them difficult to process using standard language models. A primary hurdle was the need for an AI model with a sufficiently large context window to ingest entire documents or large sections at once. To address this, we employed models like Gemini, selected for their extended context window capabilities. However, even these models were constrained by limited output token windows, which prevented full extraction of character data in a single pass. This necessitated segmenting the input and issuing multiple API calls per document. Initial tests with open-source models such as LLaMA and Mistral (under 32B parameters) were computationally intensive and failed to consistently capture the complex structure and semantics of scientific writing. These models exhibited high latency, low accuracy, and significant

difficulty in parsing detailed morphological tables. The scale of this project, requiring hundreds of concurrent character extractions across over 300 documents, made such models impractical without access to significant compute resources. Parallelization through concurrent API calls was essential for performance, reducing what would otherwise be hours of manual data entry to a few minutes per matrix. Although models like Gemini Pro have known memory limitations, ongoing optimization efforts aim to improve efficiency and stability, with the goal of releasing a robust, well-tested public version of the tool to support data curation through platforms like MorphoBank.

In addition to technical barriers, the integration of AI into data curation raises important ethical and methodological concerns. A key issue is bias: large language models trained on incomplete, outdated, or unrepresentative data may introduce systematic errors or misinterpret morphological information. Such biases can compromise dataset reliability and mislead downstream research. To mitigate this, we emphasize the need for validation workflows that compare AI-extracted outputs with human-curated standards. Transparent documentation of model behavior, training data limitations, and known failure modes is essential for building trust in AI-generated datasets. Furthermore, reproducibility remains a central concern. Scientific datasets must be verifiable and reusable, which requires AI tools to generate outputs that are not only accurate but also transparently produced and well-documented. To ensure broad acceptance and adoption of AI-assisted curation in systematics and paleontology, we advocate for open-access methodologies, detailed protocol sharing, and ongoing community evaluation. Only through such transparency can AI tools become a trusted component of research infrastructure and contribute meaningfully to long-term data stewardship.

Conclusions

This study demonstrates the potential of AI to transform morphological data curation by automating the extraction and structuring of character data from scientific literature. Our web-based tool generates standardized NEXUS files, significantly reducing manual effort while improving consistency and reusability. Current development focuses on improving accuracy and addressing memory limitations in large language models like Gemini Pro. Future iterations will enhance performance and support broader dataset formats. The tool will be made publicly available through MorphoBank, providing an accessible resource for researchers in paleontology, systematics, and evolutionary biology. While developed for morphological data, this approach can be extended to genomic, ecological, and functional trait datasets by supporting scalable, reproducible data curation across the life sciences.

List of Abbreviations

AI: Artificial Intelligence

FAIR: Findable, Accessible, Interoperable, and Reusable

LLM: Large Language Model

ML: Machine Learning

PBDB: PaleoBiology DataBase

PDF: Portable Document Format

API: Application Programming Interface

OCR: Optical Character Recognition

JSON: JavaScript Object Notation

Acknowledgements

This work was supported by Phoenix Bioinformatics and the US National Science Foundation (NSF-DBI-2049965 and NSF-EAR-2148768). We would like to thank Dr. Maureen A. O'Leary for her ongoing support. We also acknowledge the use of Gemini, which contributed to the development and testing of the AI curation tool.

References

1. Wilkinson MD, Dumontier M, Aalbersberg IjJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016 Mar 15;3(1):160018.

2. Hardisty AR, Ellwood ER, Nelson G, Zimkus B, Buschbom J, Addink W, et al. Digital Extended Specimens: Enabling an Extensible Network of Biodiversity Data Records as Integrated Digital Objects on the Internet. BioScience. 2022 Oct 1;72(10):978–87.

3. Webster MS, editor. The Extended Specimen: Emerging Frontiers in Collections-Based Ornithological Research. Boca Raton: CRC Press; 2017. 252 p.

4. Uhen MD, Allen B, Behboudi N, Clapham ME, Dunne E, Hendy A, et al. Paleobiology Database User Guide Version 1.0. PaleoBios [Internet]. 2023 [cited 2025 Jul 3];40(11). Available from: https://escholarship.org/uc/item/6tm05630

5. Ross SI, Martinez F, Houde S, Muller M, Weisz JD. The Programmer's Assistant: Conversational Interaction with a Large Language Model for Software Development. In: Proceedings of the 28th International Conference on Intelligent User Interfaces [Internet]. New York, NY, USA: Association for Computing Machinery; 2023 [cited 2025 Jul 3]. p. 491–514. (IUI '23). Available from: https://dl.acm.org/doi/10.1145/3581641.3584037

6. Dagdelen J, Dunn A, Lee S, Walker N, Rosen AS, Ceder G, et al. Structured information extraction from scientific text with large language models. Nat Commun. 2024 Feb 15;15(1):1418.

7. Gougherty AV, Clipp HL. Testing the reliability of an AI-based large language model to extract ecological information from the scientific literature. npj biodivers. 2024 May 16;3(1):13.

8. Bhardwaj A, Kishore S, Pandey DK. Artificial Intelligence in Biological Sciences. Life. 2022 Sep;12(9):1430.

9. He Y, Mulqueeney JM, Watt EC, Salili-James A, Barber NS, Camaiti M, et al. Opportunities and Challenges in Applying AI to Evolutionary Morphology. Integrative Organismal Biology. 2024 Jan 1;6(1):obae036.

10. Porto A, Voje KL. ML-morph: A fast, accurate and general approach for automated detection and landmarking of biological structures in images. Methods in Ecology and Evolution. 2020;11(4):500–12.

11. Weaver WN, Ng J, Laport RG. LeafMachine: Using machine learning to automate leaf trait extraction from digitized herbarium specimens. Applications in Plant Sciences. 2020;8(6):e11367.

12. Younis S, Weiland C, Hoehndorf R, Dressler S, Hickler T, Seeger B, et al. Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. Botany Letters. 2018 Oct 2;165(3–4):377–83.

13. Python [Internet]. Python Software Foundation; 2024. Available from: https://www.python.org/downloads/release/python-31210/

14. PyMuPDF: Python bindings for MuPDF – a lightweight PDF and XPS viewer [Internet]. Artifex Software Inc.; 2025. Available from: https://pymupdf.readthedocs.io/

15. python-docx: create and update Microsoft Word .docx files [Internet]. Canny S. 2024. Available from: https://python-docx.readthedocs.io/

16. Pandoc: universal document converter [Internet]. Boisgerault N. 2024. Available from: https://pandoc.org/

17. LlamaParse: AI-native document parsing for LLM applications [Internet]. LlamaIndex; 2025. Available from: https://docs.llamaindex.ai/en/stable/llama_cloud/llama_parse/

18. Gemini: our most capable and general model [Internet]. Google; 2025. Available from: https://deepmind.google/models/gemini

19. Stenhouse A, Fisher N, Lepschi B, Schmidt-Lebuhn A, Rodriguez J, Turco F, et al. A vision of human–AI collaboration for enhanced biological collection curation and research. *BioScience*. 2025 Mar 28;biaf021.

20. Ross SI, Martinez F, Houde S, Muller M, Weisz JD. The Programmer's Assistant: Conversational Interaction with a Large Language Model for Software Development. In: *Proceedings of the 28th International Conference on Intelligent User Interfaces* [Internet]. New York, NY, USA: Association for Computing Machinery; 2023 [cited 2025 Jul 3]. p. 491–514. (IUI '23). Available from: https://dl.acm.org/doi/10.1145/3581641.3584037

21. Dagdelen J, Dunn A, Lee S, Walker N, Rosen AS, Ceder G, et al. Structured information extraction from scientific text with large language models. *Nat Commun*. 2024 Feb 15;15(1):1418.

22. Gougherty AV, Clipp HL. Testing the reliability of an AI-based large language model to extract ecological information from the scientific literature. *npj Biodivers*. 2024 May 16;3(1):13.

23. Weaver WN, Ng J, Laport RG. LeafMachine: Using machine learning to automate leaf trait extraction from digitized herbarium specimens. *Applications in Plant Sciences*. 2020;8(6):e11367.

24. Younis S, Weiland C, Hoehndorf R, Dressler S, Hickler T, Seeger B, et al. Taxon and trait recognition from digitized herbarium specimens using deep convolutional neural networks. *Botany Letters*. 2018 Oct 2;165(3–4):377–83.

25. Abdelmageed N, Löffler F, Feddoul L, Algergawy A, Samuel S, Gaikwad J, et al. BiodivNERE: Gold standard corpora for named entity recognition and relation extraction in the biodiversity domain. *Biodiversity Data Journal*. 2022 Oct 7;10:e89481.

26. Webster MS, editor. *The Extended Specimen: Emerging Frontiers in Collections-Based Ornithological Research*. Boca Raton: CRC Press; 2017. 252 p.

27. Hammond, L., Kunzler, B., Loke, K., Mohamed, A., & Paic, A. (2025). AI-powered immediate response to pandemics: Summaries of top initiatives. OECD AI Policy Observatory.

28. Hardisty AR, Ellwood ER, Nelson G, Zimkus B, Buschbom J, Addink W, et al. Digital Extended Specimens: Enabling an Extensible Network of Biodiversity Data Records as Integrated Digital Objects on the Internet. *BioScience*. 2022 Oct 1;72(10):978–87.

29. Lannom L, Koureas D, Hardisty AR. FAIR Data and Services in Biodiversity Science and Geoscience. *DI*. 2024 Aug 13;2(1–2):122–30.

30. O'Leary MA, Kaufman S. MorphoBank: phylophenomics in the "cloud." *Cladistics*. 2011;27(5):529–37.