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, manual curation of datasets is 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-assisted curation tool developed for MorphoBank, an open access repository established to enhance standardization and usability of morphological character datasets. Specifically, this work presents an AI 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. The system enables efficient conversion of matrix-only files into complete, machine- and human-readable datasets that include key character metadata. By automating these tasks, the tool significantly accelerates dataset curation while improving accuracy and standardization. This approach increases the FAIRness of the data and offers a scalable framework for extending AI-assisted curation to standardize other biological datasets. Our findings demonstrate the value of AI in scientific data curation and advancing data reuse in paleontology, systematics, and evolutionary biology.

Introduction

Artificial intelligence (AI) is transforming scientific research across disciplines, including how data is curated, validated, and reused. While the challenge of working with high data complexity is shared across paleontology and biology (and all other fields of science), different subfields often work with fundamentally distinct data types (such as environmental settings, taxonomic occurrence records, and morphological trait descriptions), each with its own formatting standards and curation needs. Morphological datasets, in particular, are frequently encoded in inconsistent, non-standardized formats or embedded in legacy literature, making them especially difficult to extract, structure, and reuse. AI tools can streamline curation by automating data extraction and standardization tasks, reducing human error, and improving compliance with Findable, Accessible, Interoperable, and Reusable (FAIR) data principles [1].

AI is becoming increasingly common in research workflows. Large Language Models (LLMs) and other AI-powered tools are already assisting with tasks such as drafting scientific manuscripts, generating code, and annotating data [2-5]. While AI is well-established in fields like drug development and genomics [6, 7], its application to evolutionary and ecological research (such as automated detection of morphometric landmarks and the conceptualization of AI-based digital curators for biological collections) is more recent [8-10]. These emerging approaches suggest a critical role for AI in supporting scalable and standardized curation of complex datasets in paleontology, biology, and other scientific fields.

This study builds on these approaches by implementing an AI-assisted pipeline for curating morphological character matrices from published literature, designed to facilitate reuse in

open-access repositories, such as MorphoBank (www.morphobank.org). The system leverages Machine Learning (ML) models to extract morphological character names and character states from scientific publications and integrates the character information into a structured NEXUS file format for phylogenetic analyses. By reducing the time and effort required and improving accuracy and standardization, the developed tool offers a powerful solution for increasing the FAIRness of morphological datasets.

Morphological datasets are foundational to biology and paleontology, encoding traits (characters) and their observed variation (character states) across taxa, and are often visualized as a matrix of rows (characters) and columns (character states) where the value of each matrix element represents the state of the character of the given taxa. However, morphological character matrix curation is a time-intensive and error-prone process. Many morphological character matrices exist only within published literature, often embedded as tables within the paper. Researchers must often manually extract character descriptions from publications, standardize terminology, and reformat data into structures files such as NEXUS [11] or TNT. Inconsistencies in character definitions, formatting, and missing data make this task particularly challenging and susceptible to transcription errors. Scalable, automated curation tools are urgently needed to close these gaps.

The novel curation tool presented here illustrates AI's potential to close the gap in reliable, robust, automated data curation technologies by supporting large-scale paleontological and biological data curation and offering a framework for expanding AI-assisted curation approaches to other subdisciplines. This AI-assisted curation tool is currently implemented at MorphoBank,

a digital repository for morphological data that facilitates real-time collaboration and dataset archiving. In collaboration with the Paleobiology Database (PBDB) [12], 595 NEXUS files were transferred to MorphoBank. However, these files lacked critical metadata (including the vital and curation effort-intensive CHARACTER blocks), making the data difficult to interpret and reuse without referring to the original publications. 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 took upwards of two hours to curate manually. The AI-assisted curation 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.

To increase efficiency in adding character names and character states to the NEXUS files, a web-based AI tool was developed to automate the extraction and integration of character information to NEXUS files for deposition as FAIR data in MorphoBank. The AI-assisted curation tool extracts 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. This implementation demonstrates the utility of AI for large-scale data curation and sets a precedent for extending similar tools to other biological data types and repositories.

Materials and Methods

This study developed an AI-assisted curation tool, called MatrixCurator, to automate the extraction and structuring of morphological character data from scientific literature into the community-standard NEXUS file format. The tool uses an LLM and a modular pipeline that

includes preprocessing, LLM-assisted data retrieval and validation, structured schema conversion, and performance evaluation (Fig. 1). The workflow begins with user inputs: a file (e.g., .docx, .pdf) representing part or all of the journal article or supplemental data, a .nex file that contains only the data matrix and taxon information, character count specifications, the relevant page range, and the selection of an appropriate LLM document parser. The relevant pages of the article are then spliced and parsed for data extraction. A Retriever Agent extracts the characters data as a JSON (JavaScript Object Notation) object. The structured data, now a JSON object, undergo two validation steps: 1) a Count Evaluator that checks if the extracted content adheres to character count specifications, and 2) a Data Evaluator that 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. The pipeline is implemented as a web-based app and has been integrated into workflows for morphological character matrix curation in MorphoBank.

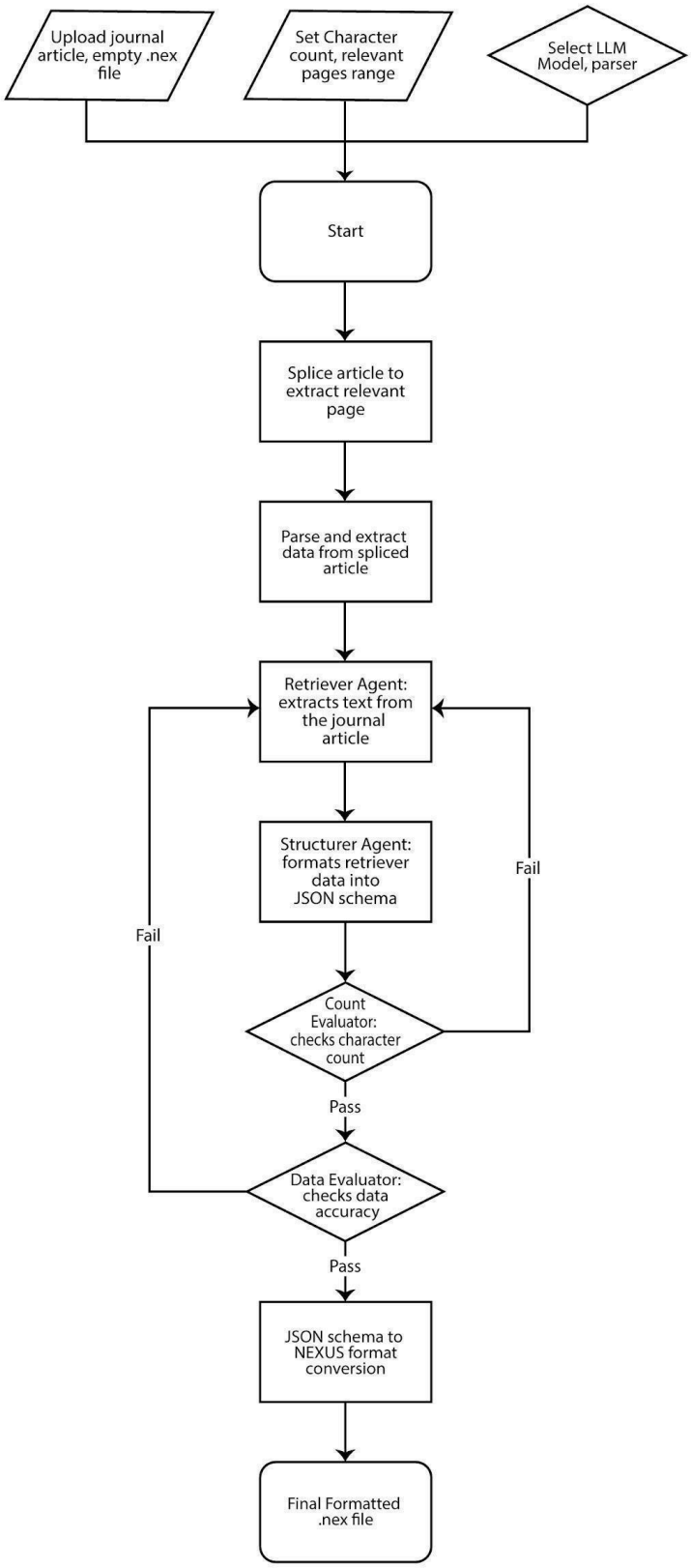
Upload journal article, empty .nex file
Set Character count, relevant pages range
Select LLM Model, parser
Start
Splice article to extract relevant page
Parse and extract data from spliced article
Retriever Agent: extracts text from the journal article
Structurer Agent: formats retriever data into JSON 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.

NEXUS File Format

The NEXUS file format is a standard widely used in systematics and phylogenetics to encode different types of data [11]. It consists of structured blocks of data including (but not limited to) TAXA, CHARACTERS, MATRIX, and TREES. The TAXA block lists the organisms used in the research, while the CHARACTERS block defines the morphological traits or features being compared amongst the organisms in the TAXA block. The MATRIX block contains the character state scores for each taxon in the dataset. These values are essential for phylogenetic inference. While not all blocks are required to run an analysis, the CHARACTERS and MATRIX blocks are essential for morphological character data analyses; they encode both the structure and content of the dataset. An optional but highly useful feature, the STATELABELS command within the CHARACTERS block, allows researchers to define each character and the character states. This feature is critical for making morphological datasets FAIR, as well as providing reusable data to enable replicable, reproducible, and transparent research. The STATELABELS command lists each character by number, followed by a label describing the character, and, if desired, a list of possible states for that character. The AI-assisted pipeline described in this paper was developed to reconstruct these missing blocks by extracting and standardizing the character descriptions from the original publications. In the example shown in Box 1, STATELABELS defines the meaning of each binary-coded character and the MATRIX block encodes which state (0 or 1) each taxon possesses. This structure of the NEXUS file is both human- and

machine-readable, and nearly all phylogenetic analysis software available today are set up to ingest and analyze data in this standard format,

Box 1. Encoding in NEXUS Format
This truncated example demonstrates how a morphological character matrix from MorphoBank P4896 (http://dx.doi.org/10.7934/P4896) is encoded in NEXUS format.

```
#NEXUS

BEGIN TAXA;
	TITLE Taxa;
	DIMENSIONS NTAX=3;
	TAXLABELS
		Afrolucina_lens Anodonia_alba Armimiltha_disciformis
	;

END;

BEGIN CHARACTERS;
	TITLE  Character_Matrix;
	DIMENSIONS  NCHAR=2;
	FORMAT DATATYPE = STANDARD GAP = SYMBOLS = " 0 1 2 3";
	CHARLABELS
		 [1] 'Overall shell shape'
		 [2] 'Maximum adult length'
;

	STATELABELS
		1
		'subcircular'
		'ovate'
		'obliquely ovate'
		'subtrigonal'
		,
		2
		'<5 mm (very small)'
		'5-24 mm (small)'
		'25-49 mm (medium)'
		'50-86 mm (large)'
		'>86 mm (very large)'
;

	MATRIX
	Afrolucina_lens		01
	Anodonia_alba		 03
```

```
        Armimiltha_disciformis     13
;

END;
```

Data Acquisition and Preprocessing

This study uses an AI-assisted curation tool to extract morphological character data from scientific publications and add these data to a NEXUS-compliant file for incorporation 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 which 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 and is described as follows.

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. The initial step in the preprocessing pipeline involves splitting the document into individual pages containing the identified relevant data using code written in 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 Python package PyPDF2 [14] and the open-source Libre Office [15] suite, 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 Libre Office [15] 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 an 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 using pypdf [16] before being processed by Gemini's Native Vision.

Large Language Models

As previously mentioned, this study leverages the Gemini family of LLMs from Google AI, specifically gemini-2.5-flash-lite, gemini-2.5-flash, gemini-2.5-pro, gemma3:27b and llama4:16x17b. 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. According to Google, 1 million tokens 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.

Our methodology employs a multi-agent (i.e., a system involving multiple AI entities working collaboratively to complete complex tasks) approach. The two distinct agents in our system are the Retriever, designed to extract information about each character as a JSON object; and the Evaluator, responsible for assessing the quality and accuracy of the extracted data. Due to the faster inference speeds and lower cost, the “flash” Gemini models were selected for use in the Retriever agent. The high-speed requirements of the agent, 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. 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 to the original paper. 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. Specifically, the Evaluator receives the original prompt input, the LLM's initial response prediction, and the parsed paper content reference. The Evaluator analyzes these elements and assigns a numerical score (1-10) reflecting the degree of correctness of the model's output. This score is then extracted from the evaluation results and returned for further analysis. This evaluation is performed using LangChain's Scoring Evaluator, which is configured with the correctness criteria. The Evaluator uses a correctness-based scoring function using a preconfigured tool called "labeled_criteria", which is loaded with the specified LLM.

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. It then caches the input tokens, and refers 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 enable systematic computational analysis, the LLM is instructed to generate the data as a JSON using a predefined JSON schema. JSON is a widely adopted, language-independent data interchange format that provides a user-friendly, human-readable structure. Its ubiquity across programming languages and compatibility with modern AI APIs make it an ideal standard for structuring and processing model outputs [19, 20]. We used JSON 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); this conversion was necessary to enable the use of phylogenetic software packages for further analysis. Each character and its associated states was 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 by 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 the initial testing phase, it was recognized that the Retriever would benefit from the “pro” model enhanced capabilities in handling more complex retrieval tasks. Complementing the Evaluator Agent was the Count Evaluator routine which 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 time and resource-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/tair/matrixcurator). The correct branch was selected, and streamlit_app.py was designated as the main file. Necessary application information, 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 the terminal command 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.

Data Extraction Performance and Efficiency

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 (Supporting Information and https://doi.org/10.5281/zenodo.17291372), specifically selected because they contain morphological matrices and represent a range of formatting complexities and layout styles. The benchmark compares a human-generated 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 text inputs are.

Agent Performance and Efficiency

To measure the performance and efficiency of our multi-agent system, we established a benchmark using a dataset of 32 papers published between 1991 and 2020 (Supporting Information and https://doi.org/10.5281/zenodo.17291372). This collection was curated to represent a diverse range of difficulties. The model's multimodal capabilities were leveraged to process the article files in PDF and DOCX formats were parsed into plain text using LibreOffice.

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 and open source models. The open source models were benchmarked on a single NVIDIA A100 80GB GPU, with costs calculated based on an hourly rental rate of $6.25. Notably, the evaluation step was omitted for the open source models, as they consistently returned a perfect value of 10 during testing. Therefore, the reported cost and inference time for these models do not include data from this step.

Results

Results of the data extraction performance analysis are given as average similarity ratios for each parsing module in Table 1. Gemini Native Vision achieved the highest average similarity ratio (0.86), followed by PyMuPDF (0.66), Pandoc (0.59), and LlamaParse (0.46). Table 2 contains the results of the agent performance analysis, specifically the average inference time and cost for each agent using different models. Table 3 shows the average accuracy and success rate for each agent using different Gemini models and open source models.

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

Table 2: Average inference time and cost for each agent using different Gemini models and open source models.

Retriever	Evaluator	Average Inference Time (s)	Cost per Character (USD, $)
gemini-2.5-flash	gemini-2.5-pro	1.3530	0.0035
gemini-2.5-flash-lite	gemini-2.5-flash	1.1780	0.0011
gemini-2.5-flash-lite	gemini-2.5-pro	1.9581	0.0025
gemma3:27b		6.056	0.01051
llama4:16x17b		13.474	0.0233

Table 3: Average accuracy and success rate for each agent using different Gemini models and open source models

Extractor	Evaluator	Success Rate	Average Accuracy
gemini-2.5-flash	gemini-2.5-pro	99.95	90.91
gemini-2.5-flash-lite	gemini-2.5-flash	96.82	85.43
gemini-2.5-flash-lite	gemini-2.5-pro	95.06	85.03
gemma3:27b		99.81	30.06

llama4:16x17b		97.54	25.09

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-assisted curation tool to over 400 published papers containing morphological character matrices (Available at 10.5281/zenodo.17291371). 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 system successfully extracted and structured morphological character data from diverse formatting styles, table structures, and text descriptions. Despite substantial

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. In curator review, outputs often required only light editing such as correcting punctuation and minor formatting errors.. More substantial intervention was occasionally needed when source materials were unusually complex or inconsistent. Even in these cases, starting from AI-generated output was considerably faster than beginning transcription from scratch, enabling a substantial reduction in the manual effort needed for dataset preparation. These experiences suggest that the tool can play a valuable role in accelerating morphological data reuse, while still relying on human expertise to ensure accuracy and resolve more challenging cases. By combining automation with curator oversight, the system helps fill a critical gap in supporting large-scale, reproducible phylogenetic research.

Discussion

This study shows that AI-assisted methods can substantially reduce the effort required to curate morphological matrices while improving the completeness and standardization of character–state information for reuse. By extracting character labels and states from publications and assembling NEXUS files suitable for MorphoBank, the workflow shifts curator time from transcription to quality control and advances the reusability and interoperability of morphological datasets across phylogenetics, systematics, and comparative biology [1,30–32].

Model and parser choice proved pivotal to quality of data curation. Gemini Native Vision produced the most faithful text recovery from PDFs and DOCX, easing downstream extraction, and an agent configuration using Gemini 2.5 Pro for evaluation yielded the highest average

accuracy (90.91%). Faster “flash” variants were more economical but showed a modest accuracy drop into the mid-80% range, reflecting a practical speed-cost-accuracy trade-off. Context caching reduced token use by roughly ninety percent on dense matrices, enabling repository-scale processing without prohibitive expense. In practice, curators typically made minor logistical edits (punctuation and formatting), with deeper intervention only for unusually complex layouts of the data in their original form (e.g., journal article).

These contributions support FAIR data principles. By extracting and standardizing character state definitions into CHARSTATELABELS blocks and depositing enriched NEXUS files in MorphoBank, the tool improves dataset accessibility, interoperability, and reusability. Standardized, machine-readable formats (NEXUS with JSON intermediates) enable indexing by trait, taxon, or project; integration with tools such as TNT, PAUP*, and MrBayes; and transparent provenance for replication and extension [1,30–32]. As a result, tasks that previously required hours per matrix can be completed in minutes plus targeted review, making large and legacy datasets substantially more accessible.

Benchmark analyses also revealed a marked disparity for open-source models. Gemma and Llama reported very high “success rates” (99.81% and 97.54%) yet exhibited low factual accuracy (30.06% and 25.09%), indicating a tendency toward confident but incorrect outputs. They were also slower and more expensive per character than the top-performing Gemini configuration, making them impractical for reliable, scalable deployment in this workflow at present. Limitations remain and motivate curator-led verification. We observed omissions in dense or irregular tables, misalignments when headings repeat, occasional fabricated states,

ambiguous anatomical terms resolved incorrectly, and routine normalization issues. These patterns underscore the need for explicit validation workflows that compare extracted outputs with source text, parser and prompt safeguards, curator checklists, and transparent documentation of model behavior and known failure modes [22,23]. Bias in training data can introduce systematic errors; clear reporting of constraints and ongoing evaluation are essential for trust.

Beyond immediate automation, this work aligns with emerging visions of AI "digital curators" that augment the efforts of expert, human curators in task-specific collaboration. Such agents can be orchestrated for document parsing, metadata enrichment, anomaly detection, and hypothesis generation, adapting to curatorial goals and integrating with repository workflows [9,21–23,28]. Separate, complementary efforts demonstrate automated trait extraction from images and multimedia, taxon and phenotype recognition, and structured biodiversity text mining [24–27]. Linking these capabilities supports extended digital specimen networks and other integrated infrastructures that enable a more holistic view of biodiversity [29], consistent with MorphoBank's mission and platform [31].

The practical impact of AI-assisted curation in general is evident, as the results from the novel tool developed here show. Manually curating a 120-character matrix typically takes about two hours but our tool generates an equivalent file in under 1 minute, with quality control completed in roughly twenty minutes. This efficiency reduces labor, minimizes human error, and streamlines updates such as adding new taxa with consistent character-state encoding. 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. This speed of curation using our novel AI-assisted curation tool directly supports long-term usability across databases and disciplines and strengthens repository support for large, heterogeneous datasets [30–33].

Conclusions

This study demonstrates that AI can materially transform morphological data curation by automating the extraction and structuring of character data from scientific literature. Our web-based, AI-assisted tool assembles standardized NEXUS files, markedly reducing manual effort while improving consistency and reusability. Ongoing development targets accuracy and context-window constraints in large language models (e.g., Gemini Pro), with planned enhancements for performance and broader dataset formats. We intend to make the tool publicly available through MorphoBank, providing an accessible resource for researchers in paleontology, systematics, and evolutionary biology. Although built around morphological matrices, the approach generalizes to genomic, ecological, and functional trait datasets, supporting scalable, reproducible curation across the life sciences [1,30–32].

Beyond automation, this work aligns with a growing vision of AI “digital curators” that augment expert practice through adaptive, task-specific collaboration. Such agents, ranging from conversational assistants trained on collection-specific knowledge to specialized components for information extraction and metadata enrichment, can also support anomaly detection and

hypothesis generation, integrate trait data from images and multimedia, and flag inconsistencies in legacy datasets for curator review [9,21–23,24–27]. Selecting and orchestrating appropriate agents can itself be AI-assisted, enabling a flexible ecosystem that links disparate datasets and surfaces overlooked patterns, thereby contributing to extended digital specimen networks and integrated digital infrastructures that offer a more holistic view of biodiversity [28,29]. This vision is consonant with MorphoBank's mission, where standardization, accessibility, interoperability, and reusability are foundational; the system presented here is a concrete step toward that future, indicating how the right infrastructure and human–AI interaction design can make curatorial workflows faster, more accurate, and more adaptable to modern biodiversity research [31].

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

Data Availability

All relevant data are within the manuscript and its supporting Information files. Code to execute MatrixCurator can be found at https://github.com/tair/matrixcurator and lists of projects whose matrices were used for AI training are in Supporting Information and available at https://doi.org/10.5281/zenodo.17291372

Acknowledgements

This work was supported by Phoenix Bioinformatics and the US National Science Foundation (NSF-DBI-2049965 and NSF-EAR-2148768). This project would not have been possible without the support of the Phoenix Bioinformatics team (in alphabetical by first name order): Alyssa Proia, Amina Khababa, Connie Ng, Eileen Conroy, Erika Bakker, Josh Young, Kartik Khosa, Leonore Reiser, Swapnil Sawant, Trilok Prithvi, and Xingguo Chen. We are grateful for Dr. Maureen A. O'Leary's continued guidance on MorphoBank. The development and evaluation of the AI-assisted curation tool described here involved the use of multiple LLMs, including Gemini (Google DeepMind), LLaMA the Gemini API and LlamaCloud platforms to assist with document parsing, character-state extraction, and structured output generation. The authors also acknowledge the use of OpenAI's ChatGPT as a writing assistant during the preparation of this manuscript. ChatGPT (GPT-5) was used to help improve text clarity, structure, grammar, and formatting. All intellectual content, research design, and conclusions are the work of the authors.

References

1. Wilkinson MD, Dumontier M, Aalbersberg IJ, 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. Donato H. Artificial Intelligence Revolutionizing Scientific Publishing. Internal Medicine [Internet]. 2024 May 17 [cited 2025 July 3];31(1-Edição Especial). Available from: https://revista.spmi.pt/index.php/rpmi/article/view/2591

3. Pividori M. Chatbots in science: What can ChatGPT do for you? Nature [Internet]. 2024 Aug 14 [cited 2025 July 3]; Available from: https://www.nature.com/articles/d41586-024-02630-z

4. Ajilore OA. When ChatGPT Met RDoC: Leveraging Artificial Intelligence to Bridge the Gap Between Data and Prognosis. Biological Psychiatry. 2024 Dec 15;96(12):902–3.

5. von Garrel J, Mayer J. Artificial Intelligence in studies—use of ChatGPT and AI-based tools among students in Germany. Humanit Soc Sci Commun. 2023 Nov 9;10(1):799.

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

7. Bijli MK, Nisa UU, Makhdomi AA, Hamadani H. Chapter 2 - The synergy of AI and biology: A transformative partnership. In: Hamadani A, Ganai NA, Hamadani H, Bashir J, editors. A

Biologist's Guide to Artificial Intelligence [Internet]. Academic Press; 2024 [cited 2025 July 3]. p. 13–34. Available from: https://www.sciencedirect.com/science/article/pii/B9780443240010000026

8. 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.

9. 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.

10. 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.

11. Maddison DR, Swofford DL, Maddison WP. Nexus: An Extensible File Format for Systematic Information. Systematic Biology. 1997 Dec 1;46(4):590–621.

12. 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 July 3];40(11). Available from: https://escholarship.org/uc/item/6tm05630

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. LibreOffice: a free and open-source office productivity software suite [Internet]. The Document Foundation; 2011. Available from: https://www.libreoffice.org/

16. pypdf: a free and open-source pure-python PDF library [Internet]. Fenniak M. 2006. Available from: https://pypdf.readthedocs.io/

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. Geng S, Cooper H, Moskal M, Jenkins S, Berman J, Ranchin N, et al. JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models [Internet]. arXiv; 2025 [cited 2025 Aug 27]. Available from: http://arxiv.org/abs/2501.10868

20. Neubauer F, Pleiss J, Uekermann B. AI-assisted JSON Schema Creation and Mapping [Internet]. arXiv; 2025 [cited 2025 Aug 27]. Available from: http://arxiv.org/abs/2508.05192

21. 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

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

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

28. 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.

29. 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.

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

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

32. Long-Fox BL, Andruchow-Colombo A, Jariwala S, O'Leary MA, Berardini TZ. Addressing Global Biodiversity Challenges: Ensuring Long-Term Sustainability of Morphological Data Collection and Reuse through MorphoBank. In: Biodiversity Information Science and Standards [Internet]. Pensoft Publishers; 2024 [cited 2025 Aug 28]. p. e135124. Available from: https://biss.pensoft.net/article/135124/

33. Long-Fox BL. A morphological character matrix describing shells from extant and extinct lucinid bivalves of the Western Atlantic. Historical Biology. 0(0):1–10.