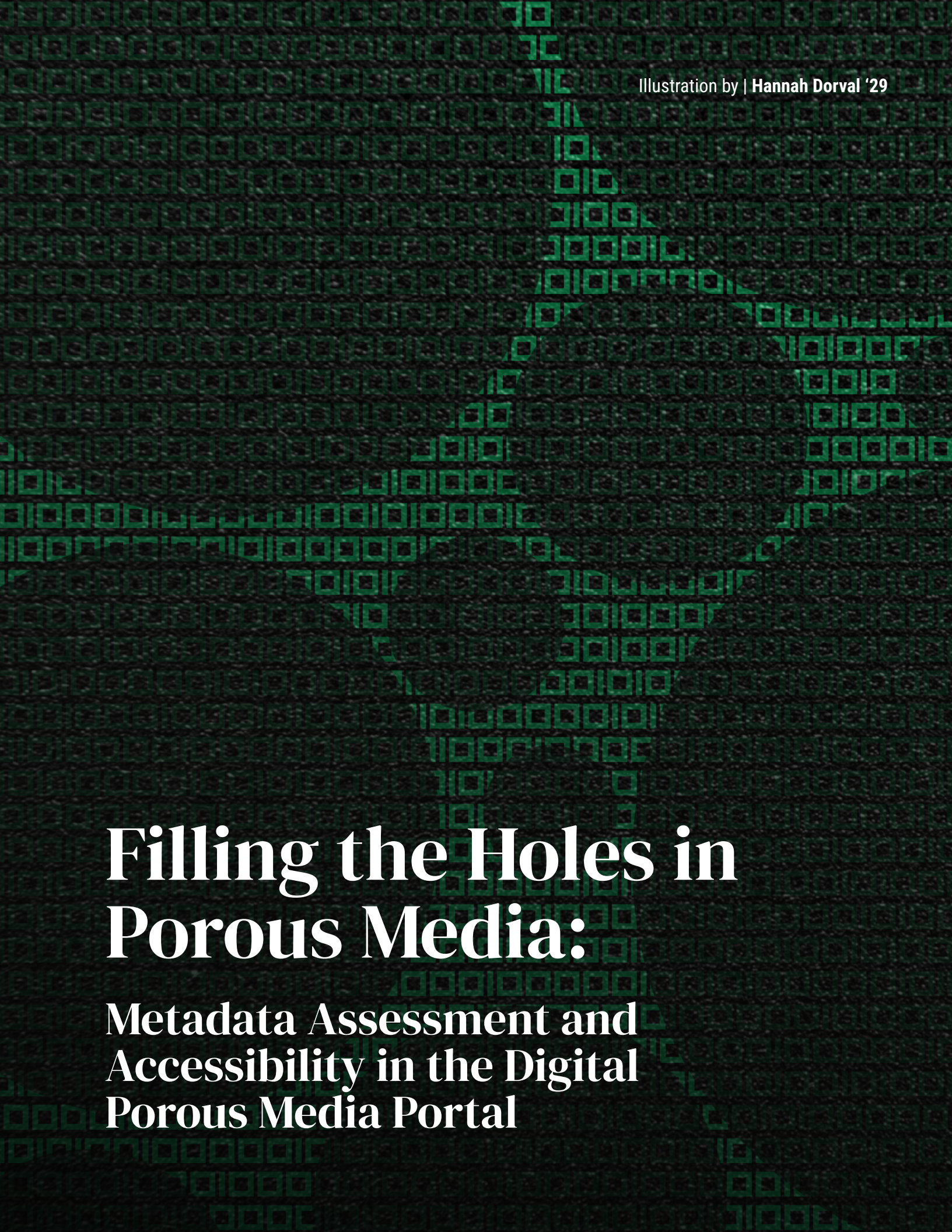


Illustration by | **Hannah Dorval '29**

Filling the Holes in Porous Media:

Metadata Assessment and Accessibility in the Digital Porous Media Portal

ORIGINAL RESEARCH

Filling the Holes in Porous Media: Metadata Assessment and Accessibility in the Digital Porous Media Portal

Zach Nowacek '27, Maria Esteva, Bernard Chang, Masa Prodanovic
Washington University in St. Louis, The University of Texas at Austin

The Digital Porous Media Portal (DPMP) is a repository of 176 datasets and terabytes of data. The applications of its data are scattered across a number of disciplines and areas of ongoing research including but not limited to geothermal engineering and aquifer mapping. Operators of the DPMP intend to make datasets easily accessible and searchable to facilitate data reuse. But understanding to what extent the DPMP is accessible requires knowledge of the metadata not initially available. This paper outlines the development of three metadata curation tools for the DPMP. The first of these replicates a tool used by other repositories to simply assess whether and which metadata fields are filled by each dataset. Recognizing the lack of nuance provided by this tool, another curation tool, backed by a Large Language Model, was created to grade and provide feedback on the content of dataset descriptions instead of merely availability. Having assessed metadata quality, the final tool—an AI chatbot (Rocco) backed by a graph database—was constructed to provide users with the ability to search the repository using natural language queries.

1. Introduction

Data reuse is a critical practice, allowing for the conservation of limited research resources. But it is only possible for those who know where to find the data. This makes creating repositories that centralize datasets for a particular domain a necessary step in achieving accessibility. One such repository is the Digital Porous Media Portal (DPMP). Over ten years, the DPMP has accepted 176 datasets, containing everything from fluid flow simulations to high-definition rock scans. It makes all data, no matter file size, accessible to download by anyone, as long as they sign up for a free account. Many portals charge for such a service but the DPMP does not, in the hope of encouraging reuse of the data it contains. But simply storing the data is not enough to adequately facilitate reuse. This paper details the development of three tools to benchmark and improve metadata in the DPMP with the objective of promoting the findability and accessibility of its datasets. The first of these, outlined in Section 3, replicates a tool used by other repositories to simply assess whether and which metadata fields are filled by each dataset. Recognizing the lack of nuance provided by this tool, another curation tool detailed in Section 4, backed by a Large Language Model, was created to grade and provide feedback on the content of dataset descriptions instead of merely availability. Having assessed metadata quality, the final tool–an AI chatbot (Rocco) backed by a graph database–was constructed according to Section 5 to provide users with the ability to search the repository using natural language queries.

2. Prior Work

As datasets grow in size and complexity, more applications to automate curation tasks are being developed. Examples like the DataONE metadata assessment tool [1] and Fuji [2] focus on checking data, metadata, and repository capabilities against FAIR standards and return structured feedback for improvement. These FAIR standards (findable, accessible, interoperable, and reusable) are important guiding principles facilitating data reuse among researchers. The DPMP does not have tools like DataONE [1] or Fuji [2] available to it, which hinders reuse capabilities. But as the repository continues to grow and as it expands its focus from just rocks to porous media more generally, understanding the health of its metadata is vitally important to its continued success. Section 3 of this paper will outline the development of a similar tool for use in the DPMP.

There is also room to improve these tools beyond their current capabilities. Their greatest restriction at the moment derives from the fact that they are not equipped to make assessments of content.

Illustration by | **Min-Suh Kwak '27**

Consider any metadata written by the user, such as descriptions of the data. Tools like DataONE may track information like word count but this lends little insight into the quality of the description's content. This severely limits our understanding of the quality of the metadata by reducing it to mere presence or absence.

The recent advancements in Large Language Models (LLMs) have opened the door to potential solutions to these shortcomings. But application of LLMs to data curation tasks is still quite new. Researchers in [3] used LLMs to generate new metadata intended to help improve the findability and accessibility of datasets for users. Applications of LLMs for interoperability and reusability have been explored further in [4], which seeks to address failures of individual datasets to follow defined domain standards including data types and metadata field names. Both papers acknowledge the novelty, imperfections, and difficulties of applying LLMs to data curation tasks.

Section 4 of this paper presents prototypes for a tool which uses LLMs to assess the content of user-written metadata. This tool is similarly situated to those works [3, 4] insofar as it applies LLMs to a new task in data curation with the goal of promoting FAIR principles. This tool, using general and domain-specific guidelines, will assess descriptions of datasets by grading them against a set of predetermined guidelines and providing suggestions as to how users can address the shortcomings identified.

LLMs have also shown promise as tools to assist researchers in finding datasets. Some repositories have experimented with LLMs preprompted with the metadata files stored as .json objects [5]. This is functionally Cache Augmented Generation (CAG), in which the LLM informs its answers using all context provided by the user. In this case, that context is the repository metadata. However, current models are limited by the amount of context they can use to inform responses. The scale of many repositories, including the DPMP, is so great that the text of all metadata would far exceed this limit on context and make CAG unworkable. Retrieval Augmented Generation (RAG) may provide a solution. By first identifying the most relevant datasets and then providing only that metadata as context, the issue of scale can be addressed. This is analogous to searching Google and reading the top websites to answer your question instead of trying to read every website on the Internet.

An LLM-backed query interface would have clear utility for the DPMP, allowing researchers to navigate all DPMP metadata with simple natural language questions. But implementations like those in [5] are not scalable. These applications work for repositories of limited size but for those like the DPMP with 176 datasets, two issues arise. First, consumer products like ChatGPT will only accept so many documents, even with paid premium access. If the current load of metadata does not exceed imposed limits, it may do so soon. Second, the RAG techniques of ChatGPT are a complete black box. How these consumer-ready models work is proprietary information and is not available for inquiry or revision during the development process. Such a tool must therefore be specially fitted for the DPMP. Section 5 of this paper outlines the development of such a tool.

3. Repository-Wide Metadata Evaluation

The first tool for assessing the metadata in the DPMP is not content aware. It is a spreadsheet designed to visualize the presence or omission of metadata fields for each dataset. A tool like this works on the same premise as DataONE. It gauges simply whether or not certain fields were filled out by dataset authors. All information is displayed in a single Excel spreadsheet with each row corresponding to a dataset and each column corresponding to a metadata field of interest. When a field is filled out, that cell will be given a 1; when it is left empty it will be given a 0. If a dataset has multiple nodes of the same type, the same metadata field will be averaged across all nodes of that type. For instance, if a dataset contains three samples but only one lists the geographical location, the cell for that row would list 0.33 as can be seen in Fig. 1. For ease of understanding, the spreadsheet is also color coded. Cells with a 1 are green, those with a 0 are pink, a 0.5 is white, and values in between fall on a continuous color scale.

3.1 Methodology

The spreadsheet was assembled primarily using *openpyxl*. All the metadata for the DPMP are stored in .json files. These metadata are not the actual bulk of the data (e.g. the simulations or scans) but instead contain information about the data like titles, porous media type, and sample collection date. The information is regularly structured according to the data model outlined in [6] (see Fig. 4) and implemented in [7]. The structure is centered on a dataset node which has properties like "title" and "description." The actual data may be one of three things: a "sample," "digital dataset," or "analysis dataset." Each has its own distinguishing properties. Datasets are usually structured such that samples branch to digital datasets and digital datasets branch to analysis datasets. This can be seen in Fig. 5 where yellow nodes (samples) branch to blue nodes (digital datasets) which in turn branch to red nodes (analysis datasets). Related publications (green nodes) are also connected to many datasets.

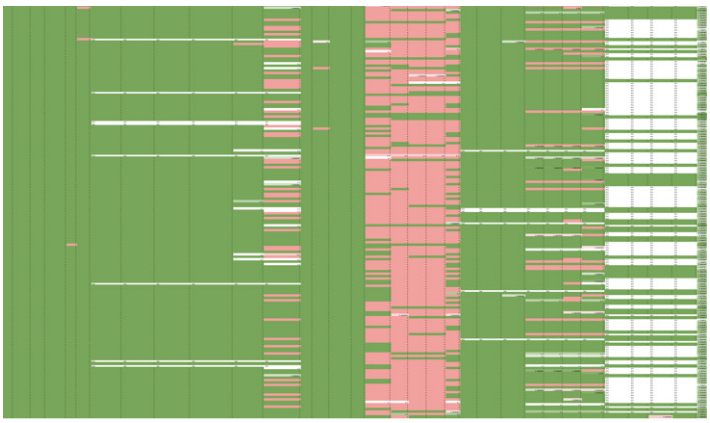


Fig. 1 | Metadata evaluation spreadsheet

Once all 176 of these files had been scraped and assembled into a single folder, each one was analyzed in turn. Because of their regular structure, traversing the objects for information of interest was merely an extensive exercise in data structures. Filtering out of some user inputs like "N/A" or "-" was necessary but mostly the content was either there or not. Because some datasets do not include some types of nodes, segments of rows were filled with "N/A" to indicate that type of node was not present in the dataset.

A "totalScore" column is also provided, which averages out all scores across the row for each dataset. It is calculated according to a user provided "requirement_weight" which affects how required fields are accounted for in the average. This is because if a field is required, it must have been filled out by the user. Every dataset therefore shares at least a baseline percentage of fields which have been completed. If a curator is curious about the completion of fields not required, "requirement_weight" may be set to zero.

3.2 Results

Fig. 1 provides a view of the metadata assessment of the entire portal. Although no particular fields may be discerned from the image, clear patterns of missing data appear, notably a block of red running top to bottom just right of center. This block corresponds with a number of sample properties including grain size and geographical location. While so much red may at first seem concerning, it would come as no surprise to most experts.

There is a reason not all fields are required, and it is because not all of them are relevant for all samples. Not all or even most samples are valuable rock cores. Some are handfuls of sand bought from a hardware store. They have no geographical location because they are thrown away after use. In other cases, properties like porosity cannot be measured while simultaneously conducting the experiment. This result, more than anything, demonstrated the limitations of a metadata assessment like DataONE's; it fails to factor in context and cannot assess content.

4. Assessing Dataset Description Conten Grading and Improving

Dataset descriptions are to datasets what abstracts are to papers. They are the first focus of attention for those looking for data. Their text is indexed and used to improve search results across different platforms, enhancing data discoverability. Ideally, they give readers a snapshot of dataset content, source, collection methodology, and purpose. But the dataset descriptions of the DPMP fall far short of this standard. The median word count is 103, too short to be meaningfully descriptive. A qualitative review of a subset of descriptions showed that many did not focus on the dataset but on the research findings, some included information already provided in other metadata fields, and others did as little as thanking contributors.

4.1 Guidelines

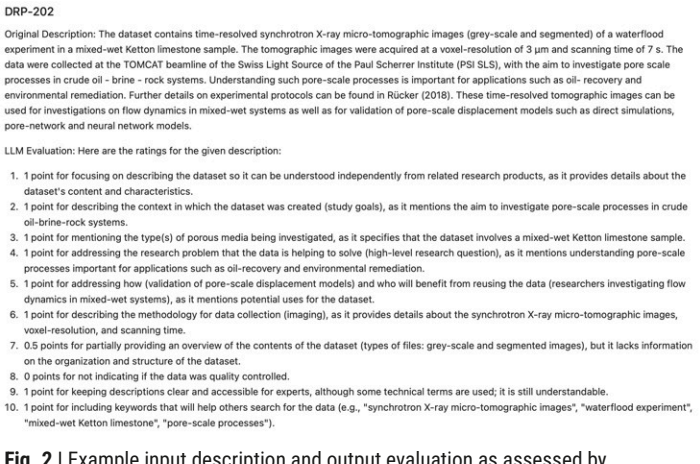
Considering this problem, we devised a set of ten norms which researchers submitting their datasets should use to evaluate and improve their descriptions. These guidelines included content requirements like the porous media involved and methodology as well as style requirements like clarity and conciseness. The guidelines were developed collaboratively by a data curator (Dr. Maria Esteva) and a domain specialist (Dr. Masa Prodanovic) to make descriptions clear, concise, and relevant to porous media. But concern that future submitters would fail to attend to these guidelines led us to develop an LLM-based tool to evaluate and improve dataset descriptions. This application focuses on content quality by embedding general dataset description best practices with those specific to porous media. This "grader" provides an evaluation of a draft dataset description by scoring it against each guideline. A prototype of the "writer" was also developed to take that evaluation and propose changes for the author's consideration.

4.2 Grading Descriptions

The grader, like most LLM applications, must be prompted in such a way so as to minimize the chances of hallucinations, crude analysis, and even simple arithmetic errors. The guardrails that we proposed for our grader consist of three main components: clear guidelines, few-shot prompting, and encouraging chain-of-thought reasoning.

One frequent strategy for AI use today is known as 'zero-shot prompting,' in which the user tasks the LLM without providing any examples. By contrast, 'few-shot prompting' guides the model by providing user-approved cases of what an input and its corresponding output should look like. Thus, when prompted with a new case, the LLM already has a structure with which to formulate its answer and will demonstrate greater competency [8]. In our case, we provided three examples of dataset descriptions which had been hand-graded by Dr. Esteva. Examples of varying quality were intentionally chosen to showcase to the LLM exemplary, middling, and poor descriptions.

The grader also takes advantage of 'chain-of-thought' (CoT) strategies. CoT prompting and reasoning have been shown to dramatically increase the reliability and accuracy of LLM responses to complex, multi-step problems across numerous models [9]. CoT reasoning decomposes problems, in this case evaluating each guideline individually, and allows the model the opportunity and space to justify its answer. This behavior can be observed under the "LLM Evaluation" heading in Fig. 2 which shows an LLM-graded description. Encouraging the model to walk through its reasoning has the added benefit of making the grader a better diagnostic tool. Were researchers to use it for their descriptions, they would know not just which guidelines were deficient but also why they were and how descriptions could potentially be improved. This CoT behavior is encouraged by the same examples which serve to few-shot prompt the LLM. These examples do not simply say that the description is insufficient in addressing a certain guideline, but why.



DRP-202

Original Description: The dataset contains time-resolved synchrotron X-ray micro-tomographic images (grey-scale and segmented) of a waterflood experiment in a mixed-wet Ketton limestone sample. The tomographic images were acquired at a voxel-resolution of 3 µm and scanning time of 7 s. The data were collected at the TOMCAT beamline of the Swiss Light Source of the Paul Scherrer Institute (PSI SLS), with the aim to investigate pore scale processes in crude oil - brine - rock systems. Understanding such pore-scale processes is important for applications such as oil- recovery and environmental remediation. Further details on experimental protocols can be found in Rücker (2018). These time-resolved tomographic images can be used for investigations on flow dynamics in mixed-wet systems as well as for validation of pore-scale displacement models such as direct simulations, pore-network and neural network models.

LLM Evaluation: Here are the ratings for the given description:

1. 1 point for focusing on describing the dataset so it can be understood independently from related research products, as it provides details about the dataset's content and characteristics.
2. 1 point for describing the context in which the dataset was created (study goals), as it mentions the aim to investigate pore-scale processes in crude oil-brine-rock systems.
3. 1 point for mentioning the type(s) of porous media being investigated, as it specifies that the dataset involves a mixed-wet Ketton limestone sample.
4. 1 point for addressing the research problem that the data is helping to solve (high-level research question), as it mentions understanding pore-scale processes important for applications such as oil-recovery and environmental remediation.
5. 1 point for addressing how (validation of pore-scale displacement models) and who will benefit from reusing the data (researchers investigating flow dynamics in mixed-wet systems), as it mentions potential uses for the dataset.
6. 1 point for describing the methodology for data collection (imaging), as it provides details about the synchrotron X-ray micro-tomographic images, voxel-resolution, and scanning time.
7. 0.5 points for partially providing an overview of the contents of the dataset (types of files: grey-scale and segmented images), but it lacks information on the organization and structure of the dataset.
8. 0 points for not indicating if the data was quality controlled.
9. 1 point for keeping descriptions clear and accessible for experts, although some technical terms are used; it is still understandable.
10. 1 point for including keywords that will help others search for the data (e.g., "synchrotron X-ray micro-tomographic images", "waterflood experiment", "mixed-wet Ketton limestone", "pore-scale processes").

Fig. 2 | Example input description and output evaluation as assessed by Llama-4-Maverick

Chain-of-thought reasoning does present a few challenges, mainly a lack of guaranteed structure. Not constricting the LLM as to how it will format and print its score makes it harder to extract those scores for each guideline and the final score overall. There are several options for collecting this information. Early iterations asked the grader to give its final score only at the end and that number was extracted using regular expression matching. This however did not allow for a breakdown by guideline. Additionally, letting the LLM add the final score together often introduced arithmetic errors. In the current version, the grader finishes an evaluation and that evaluation is passed to the LLM again. This time it is asked to output an ordered list of tuples which contain first the guideline number and second the score given in the evaluation. This output may then be parsed and used to provide an overall score as well as the full breakdown.

An evaluation of the entire portal (Fig. 3) reveals a dismal average score of only 4.65 out of 10. Just over 8.5% of descriptions scored an 8 or above, further suggesting an overall low quality.

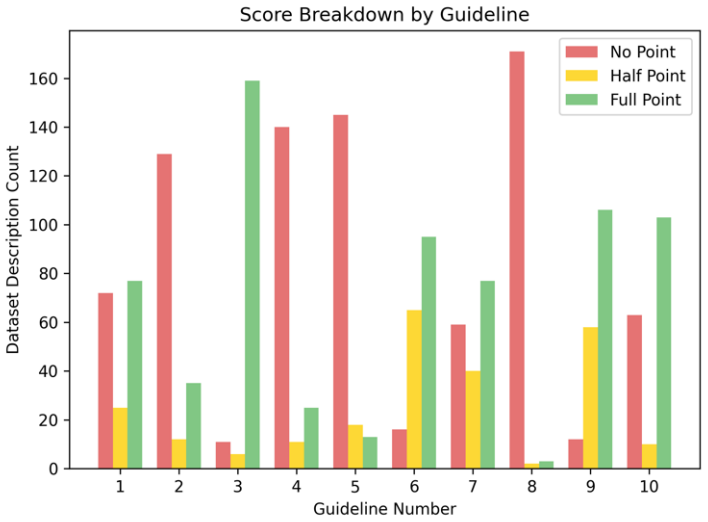


Fig. 3 | Distribution of scores in the DPMP broken out by guideline

Across descriptions, high compliance levels for guidelines 3 and 6, and low compliance for 4, 5 and 8 reveals that while most researchers describe the materials and methods used to obtain the dataset, the research questions that motivated the creation of that dataset, their potential reuse cases, and how they are quality controlled are seldom addressed. More work must be done to ensure the consistency and accuracy of the grader, ideally by collecting a sample of hand-graded examples by experts and comparing scores. Such resources were not available though at time of writing.

4.3 Improving Descriptions

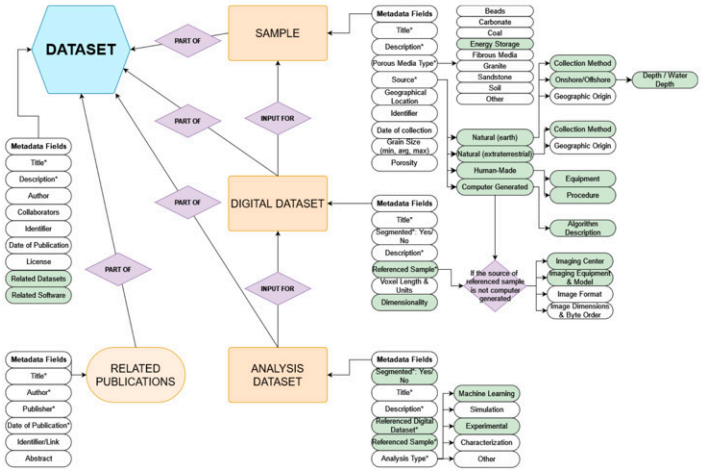


Fig. 4 | Data model created in [6]

Having diagnosed issues using the grader, the resulting evaluations can be used to inform an assistive writing tool intended to improve the quality of the dataset description with the guidance of the user. A preliminary prototype of this application has been developed to improve those dataset descriptions already present on the DPMP. In its deployable form, the application will work dynamically with the user, providing feedback and asking questions to help fill in missing information. The benefit of this system is that the model is much less likely to hallucinate or address missing information with guesses it makes from its pretrained knowledge. Instead, it will prompt users (the researchers submitting their datasets) for information which the authors will likely have access to.

In this first prototype, instead of having a researcher provide answers dynamically, we fill that gap with the text of the related papers listed by the author to inform the LLM. This serves as a proof-of-concept to demonstrate LLMs can improve descriptions against the set of guidelines provided. These papers were collected manually and then converted to strings of markdown text using the Python library *pymupdf4llm*. The LLM is instructed to take in a description, guidelines, a grader's assessment (provided by the grader), and related papers. From all of this, it is told to write a revised description, keeping original language when possible. All this information provides the LLM with what should be adequate knowledge to improve the descriptions. Early tests showed promising results on a handful of descriptions. But prompting strategies must still be refined for this static version of the writer and even more extensive work must take place to create a dynamic, deployable application.

5. Rocco, The Digital Porous Media Portal Curator

While dataset descriptions may be useful for researchers, they are only one strand in a wider web of metadata. Even perfect dataset descriptions would not contain all information the researcher requires. Researchers instead might be curious about one of the dozens of attributes of a sample, digital dataset, or analysis dataset. The difficulty of navigating those branches of a dataset can discourage researchers from finding relevant data for reuse, especially when the only search tool available is limited to precise keyword matching. Ideally, a tool would accommodate even those users with the broadest of research interests and guide them to useful datasets without prior knowledge of what the DPMP contains or how to use it.

For this purpose, we created a Digital Porous Media Portal Curator named Rocco. Rocco is a Retrieval Augmented Generation LLM application with specialized knowledge of the DPMP and its contents. From a graph database, Rocco can conduct direct queries against the metadata and semantic searching against the dataset descriptions. There is no need for the user to understand the workings of the DPMP, they may instead ask simple, natural language queries. Responses are likewise delivered in natural language. This application supports any researchers approaching the DPMP, no matter their familiarity or experience. Such accessibility lowers the barriers for data reuse dramatically.

5.1 Conversion of Raw .json Files to Neo4j Graph Database

Given the scalability concerns identified with preprompting all the .json files in a Cache Augmented Generation manner, Retrieval Augmented Generation presents advantages. Effective retrieval allows Rocco to carve an answer out of the database with a mallet and chisel instead of a jackhammer. Implementation with a graph database has clear benefits. The node and directed edge structure permits the storing of relational information. And open-source database software systems like Neo4j have worked to make integrating instances of their databases with LLMs and other code straightforward, even creating free courses with tutorials to follow. Neo4j allows users to create free, local instances of a database without restrictions. Given all these factors, we chose to use Neo4j to organize our data for implementation of our application.

The conversion from raw .json files to the Neo4j database required the use of Neo4j drivers in Python [10]. Drivers allowed the creation of nodes and relationships directly from a Jupyter Note-

book. Variables could be dynamically entered, which was vital for inputting all available information. This project was developed using a local instance of a Neo4j graph database on a virtual machine (VM) provided by the Texas Advanced Computing Center (TACC). This database stored the metadata and made it accessible for retrieval.

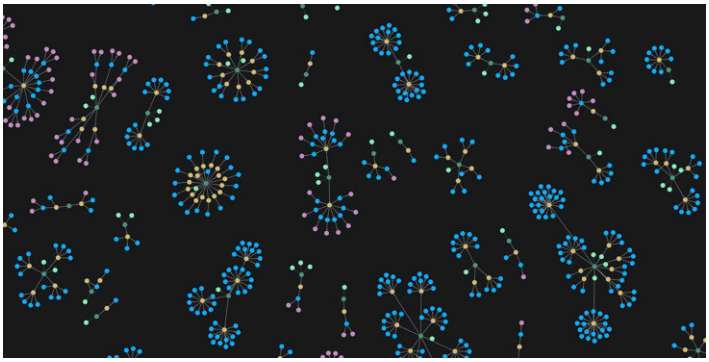


Fig. 5 | Graph database implemented in Neo4j

With the ability to parse the .json files into Python objects, the brunt of the work was traversing those objects and assigning their values to nodes and relationships in the graph database. To every extent possible, the data model shown in Fig. 4 served as a blueprint. Datasets, samples, digital datasets, analysis datasets, and related publications are all stored as nodes. Given that they contain their own information, related software and datasets will also be given their own nodes as they become a feature of the DPMP. Relationship names and direction conform to the data model and all properties listed were included. A few extra properties including file type, number of files, and dataset number were added where appropriate. A subset of the database is visualized in Fig. 5. Each isolated shape represents the metadata of a single dataset with a dataset node at the core with sample, digital dataset, analysis dataset, and related publication nodes branching out. All follow the schema laid out in Fig. 6 which is designed to emulate the data model in Fig. 4 and [6].

5.2 Rocco Formation

The models used in Rocco's construction were Meta's Llama-4-Maverick for natural language capabilities and intfloat's E5-Mistral for embedding, which were both used for their availability on the TACC's SambaNova systems. With three distinct systems to manage–Neo4j, SambaNova, and the VM–integration into one cohesive application was by far the most time intensive of all steps involved in the creation of Rocco. The structure of the application was based heavily on the Neo4j course *Building Knowledge Graphs with LLMs* [9]. The application structure is built using the libraries *LangChain*, which guides the pipeline between the LLM and the database, and *Streamlit*, which handles the user interface. Rocco has two tools from which to draw while answering user questions: natural language to Cypher query (querying) and vector similarity searching between a user query and dataset descriptions (semantic searching). Using whatever is returned by either of these searches, Rocco will report back to the user in natural language what the search found.

5.3 Querying

Neo4j graph databases operate using a language called Cypher. This was the same language used in conjunction with the driver described in Section 5.1 to create the database from .json files. Writing Cypher queries is like any coding language; it is syntactically sensitive. Without full knowledge of the database structure and labels, even users with experience coding Cypher would struggle to write queries. And users without experience would find the database completely unapproachable. The query tool allows Rocco to translate a user's question from natural language to a valid Cypher query. The database structure–including node labels, relationship types, and properties–is included. The top ten results are returned and provided as context to Rocco, who will answer the question with any relevant information. It is important to note that if the context does not contain anything relevant to a user's question, Rocco will refuse to operate off of pretrained knowledge. Instead, the user will be told that no results could be located. This step is vital to preventing hallucinations from reaching the user.

There are some refinements that are needed with this tool. The more complex an LLM-generated Cypher query, the more susceptible it is to failure. Even if correct label names and properties are called, the query structure may have faults. Queries containing UNION statements were an early example of this. However, by specifying in the system prompt how to properly conduct such queries, these issues can usually be addressed. Limiting how much information is returned by the query will also need to be addressed. At present, a query can return more information than fits in the context window which will prevent any final answer being given to the user. But too strict a limit may prevent relevant results from being returned. A dynamic system to maximize returned information without going over is therefore necessary.

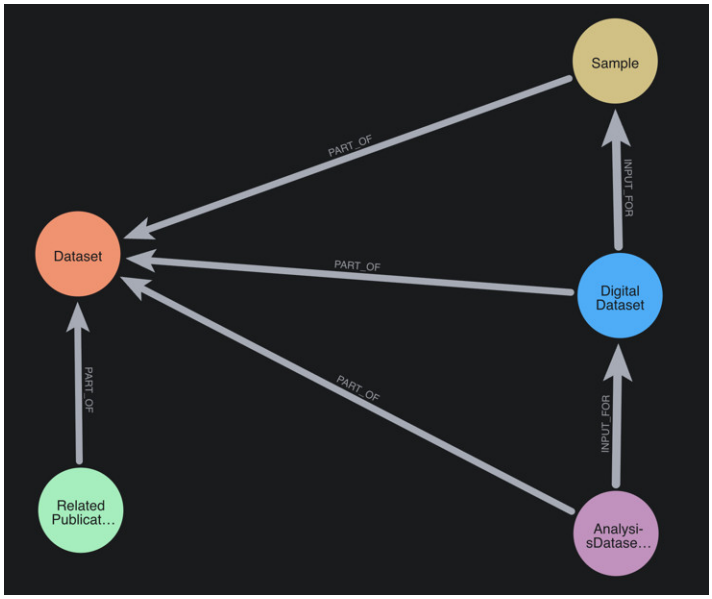


Fig. 6 | Neo4j database schema

5.4 Semantic Searching

For a user curious about finding datasets with samples that have porosities between 0.2 and 0.25, the previous tool is ideal. But researchers do not always have that specificity in mind when beginning the research process. Instead, they may have more general ideas about, for example, the kinds of simulations they want to conduct. The most straightforward way to facilitate this is to find shared keywords between the user's query and dataset descriptions. But if keywords do not match exactly, like 'sandstones' instead of 'sandstone,' the searching fails. What would be preferable is a system that matches similar *ideas* instead of simply exact *words*.

Semantic searching accommodates this goal. Text, both in the form of dataset descriptions and user queries, are embedded as vectors that capture some semantic meaning. A similarity search then locates those descriptions most similar in content to the query. While embedding queries does not currently involve preprocessing, initial attempts to embed an entire dataset description as a vector proved unreliable. Other iterations have passed the description through Meta-Llama-3.1-405B and requested a list of keywords be extracted, in an attempt to remove excessive detail and stop words. Future versions may consider experimenting with chunking the text, testing alternative models, and attempting to make the query more structurally similar to the text it seeks to match.

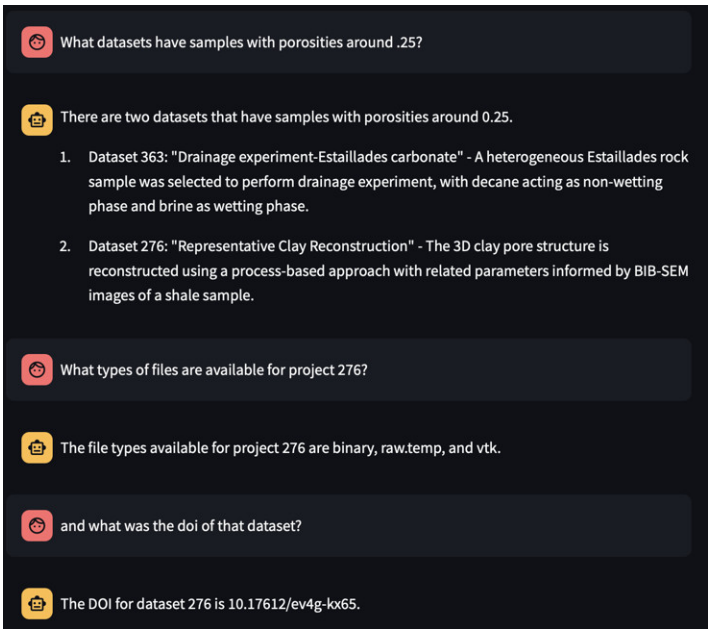


Fig. 7 | Excerpt from a conversation with Rocco

6. Conclusions

Rocco, with a fully functional user interface and search capabilities, is captured in Fig. 7. This chat exemplifies the ability of Rocco to both retrieve datasets of certain specifications and summarize their content easily for users. While not ready for deployment, the prototype is more than enough to demonstrate and explore the possibilities of a graph database RAG chatbot for the DPMP. The tool is scalable with the number of datasets and accessible for users of any technical and domain proficiency. Rocco, along with the grader and writer prototype, and spreadsheet tool outlined in this paper, push the DPMP forward in making its data more accessible. Plenty of work must still be done, including prompt refining for the grader, creation of a deployable version of the writer, and improvement of Rocco's tools (particularly semantic searching). But these curation applications, even in their early and imperfect forms, already provide a greater understanding of what data the DPMP contains and how it may be harnessed. They can be used by DPMP managers to increase the accessibility and findability of datasets and by researchers to reuse available but buried data for their own projects in pursuit of greater domain knowledge.

Acknowledgement

This paper was supported by the NSF Award #1854828, Operations & Maintenance for the Endless Frontier. The AI resources including LLM and embedding models were provided by the Texas Advanced Computing Center through their SambaNova systems. Thank you to Dr. Rosalia Gomez for her efforts coordinating this experience. Thank you to my mentors: Dr. Maria Esteva, Dr. Bernard Chang, and Dr. Masa Prodanovic. Thank you to Drs. Niall Gaffney, Gabriel Jaffe, and Vlad Krendelev for their help in working with the LLMs. Finally, thank you to Dr. Dan Stanzione for his confidence and support throughout.

References

[1] Jones, M. B., Slaughter, P., & Habermann, T. (2019). Quantifying FAIR: Automated metadata improvement and guidance in the DataONE repository network. *Zenodo*. https://doi.org/10.5281/zenodo.3408465.

[2] Devaraju, A., et al. (2020). FAIRsFAIR Data Object Assessment Metrics," *Zenodo (CERN European Organization for Nuclear Research)*. https://doi.org/10.5281/zenodo.4081213.

[3] Singh, M., Kumar, A., Donaparthi, S., & Karamelkar, G. (2025). Leveraging retrieval augmented generative LLMs for automated metadata description generation to enhance data catalogs. *arXiv*. https://doi.org/10.48550/arXiv.2503.09003.

[4] Sundaram, S. S., Solomon, B., Khatri, A., Laumas, A., Khatri, P., & Musen, M. A. (2024). Use of a Structured Knowledge Base Enhances Metadata Curation by Large Language Models. *arXiv*. https://doi.org/10.48550/arxiv.2404.05893.

[5] Zhou, X., Modak, S., Chan, Y.-C., Deng, Z., Sentis, L., & Esteva, M. (2025). Curate, Connect, Inquire: A System for Findable Accessible Interoperable and Reusable (FAIR) Human-Robot Centered Datasets. *arXiv*. https://arxiv.org/abs/2506.00220v1.

[6] Prodanović, Maša, et al. (2023). Digital Rocks Portal (Digital Porous Media): Connecting data, simulation and community. *E3S web of conferences. vol. 367, pp. 01010–01010*. https://doi.org/10.1051/e3sconf/202336701010.

[7] TACC. (2020). DRP Models.py. *GitHub*. https://github.com/TACC/Core-Portal/blob/task/digital-rocks/server/portal/apps/_custom/drp/models.py.

[8] Brown, T., et al. (2020). Language Models Are Few-Shot Learners. https://arxiv.org/pdf/2005.14165.

[9] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv*. https://doi.org/10.48550/arXiv.2201.11903.

[10] Neo4j. (2025). Neo4j Python Driver 5.28. *Neo4j.com*. https://neo4j.com/docs/api/python-driver/current/

[11] Neo4j. (2025). Building Knowledge Graphs with LLMs. *Neo4j.com*. https://graphacademy.neo4j.com/courses/llm-knowledge-graph-construction/.