Data PAL: Scientific Data Analysis through Conversational AI

Anonymous Author(s)
Affiliation
Address
email

Abstract

Scientific research requires both specialized domain knowledge and advanced data analysis skills. To support research progress and to make findings more accessible to the public, we present Data Project Analysis with Language models (Data PAL), a tool that facilitates data retrieval and analysis for large scientific data projects through conversational English.

Data PAL is designed to be adaptable to datasets from diverse fields. To evaluate its performance, we implement Data PAL for a collection of over 300,000 climate change datasets. We then crowdsource queries related to climate analysis from domain experts. We demonstrate that Data PAL's retrieved data are more relevant than baselines to user queries, with **over 20% higher accuracy** on key metrics.

1 Introduction

In many scientific domains, vast quantities of data are generated every day: astronomical observatories, weather stations and particle accelerators are just a few sources of invaluable data that serve as the backbone to modern scientific progress.

These data collection efforts culminate in large-scale *data projects* that compile information collected over time by various organizations, centered on a common scientific area. For instance, there are 13.6 million climate change datasets within the Coupled Model Intercomparison Project (CMIP) 6 [7], gathered by more than 50 institutions and comprising over 30PB of data. Similarly, the National Institutes of Health's Sequence Read Archive [11] currently contains 36PB of genomic sequencing data representing all branches of life.

Such large data collections are often organized in complex ways, stored in specialized formats, and described in esoteric terms known only to a small group of specialists. A significant bottleneck to scientific progress lies with the ability to *retrieve and analyze* the data, rather than the ability to collect the data, in many fields.

In this work, we propose **Data Project Analysis with Language models (Data PAL)**, a system to retrieve and analyze data from large scientific projects using conversational English. Data PAL will allow experts to evaluate hypotheses more quickly and increase the accessibility of these datasets to non-technical stakeholders.

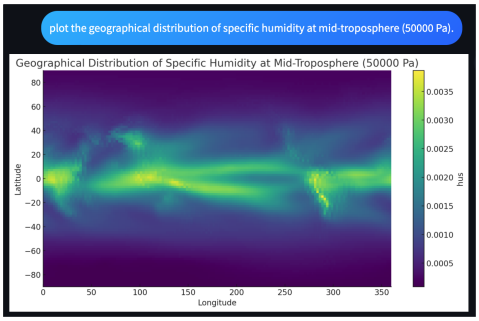

Figure 1: Data PAL allows users to engage with scientific data via conversational English through an intuitive graphical interface.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

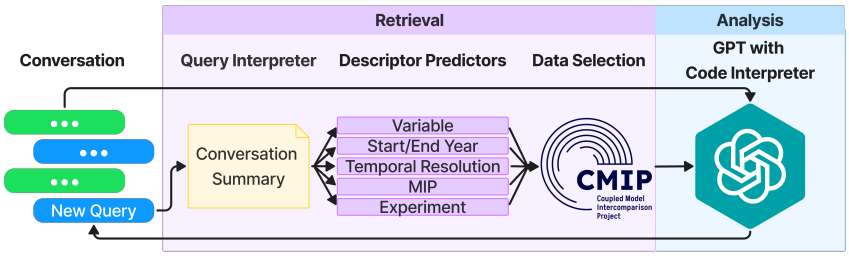

Figure 2: An overview of Data PAL. The Retrieval Component is tasked with selecting datasets to provide to the Analysis Component, based on the user query and conversational history.

Allowing the public to interact with these datasets will have important implications for education and combating scientific misinformation, by allowing users to view the data's underlying phenomena and patterns firsthand.

Data PAL's Retrieval Component employs a multi-step combination of embeddings-based and In-Context Learning (ICL) techniques to find relevant datasets, while the Analysis Component relies on the retrieved data and OpenAI's Code Interpreter Tool with GPT-4 [1, 3] to formulate a response to user queries. Our system requires no fine-tuning, which allows for novel datasets to be incorporated as a data project evolves. An example interaction with Data PAL is demonstrated in Figure 1.

As a proof of concept, we implement Data PAL for the 343,119 CMIP6 datasets generated by NASA's Goddard Institute for Space Studies (GISS). This implementation of Data PAL is, to our knowledge, the first general-use conversational system for retrieval and analysis of CMIP6 climate data. Next, we crowdsource a dataset of 35 queries from expert climate scientists, which we augment to a full dataset of 210 queries using semantic variation. Using these queries, we design a benchmark with which to evaluate Data PAL. We summarize our contributions as follows:

- We describe Data PAL, a system to enable retrieval and analysis for scientific data projects using conversational English.
- We implement Data PAL for GISS CMIP6.
- We crowdsource and manually annotate a dataset of GISS CMIP6 analysis queries for benchmarking Data PAL and similar systems in the future.
- We demonstrate Data PAL-retrieved GISS CMIP6 datasets are over 20% more accurate than baselines in several metrics.

2 Method

To motivate the design of Data PAL, we elaborate on the general structure of large data projects. Next, we discuss each component of Data PAL as summarized in Figure 2.

2.1 Data Projects

We say that a *data project* is a collection of *datasets*. Each dataset is described by a standardized set of attributes, which we refer to as *descriptors*. Common descriptors include the dataset's independent/dependent variables, the measurement units used in the dataset, or the organization that collected or owns the dataset. We assume that for each dataset accessible to Data PAL, we have access to these descriptors' *values* (it is acceptable for a dataset to have empty or "Not Applicable" values for a given descriptor). Data PAL uses the descriptor values to differentiate the datasets and select data that is relevant to the user query, as discussed in Section 2.2.

Many descriptor values consist of specialized, non-intuitive terms and abbreviations. For instance, the names of a dataset's independent and dependent variables are often highly specific and may not be well-understood by LLMs that are trained on diverse, general-purpose data. To allow Data PAL's Retrieval Component to interpret these *specialized descriptors*, we also assume access either to definitions of each possible value for each specialized descriptor, or a set of example queries and their corresponding best-match values for the specialized descriptors.

2.2 Retrieval Component

The Retrieval Component is responsible for selecting a dataset best-suited to answering the user query. Each time the user replies in a conversation, a GPT model [3, 18] is prompted to summarize the query and any conversational history into a few keywords, then determine if it is necessary to retrieve a new dataset to answer the query. If retrieval is required, the Retrieval Component first builds a profile of an ideal dataset by predicting each of the dataset's descriptor values. Next, a table of each available dataset is filtered to find the best match to these descriptors predictions.

For each descriptor, the choice of prediction algorithm is dependent on whether the descriptor is specialized, as defined in Section 2.1, or relies only on more general knowledge.

For a non-specialized descriptor, we prompt a GPT model with the conversational summary, along with ICL-style instructions detailing what information the model should extract from the summary. These instructions explain how the model should format its responses, how to handle any potential edge-cases or likely sources of prediction error, and–if available–include a few example conversational summaries with their correct descriptor value.

Specialized descriptors require additional steps to achieve accurate predictions, because GPT cannot be assumed to have adequate prior knowledge of their possible values. When a specialized descriptor's set of possible values is small (e.g., less than 10), the descriptor prediction process is similar to that of a non-specialized descriptor: we prompt GPT with the conversational summary and ICL-style instructions. Unlike non-specialized descriptors, we ensure that each possible value is defined within the ICL prompt.

For specialized descriptors with hundreds or thousands of possible values, however, there are too many definitions to fit into one prompt. In this case, we split the descriptor prediction into three steps. First, we provide GPT with the conversational summary and prompt the model to further reduce the conversation into just a few keywords likely to be pertinent to the specialized descriptor. This shortened and better-focused summary is then embedded, and the t descriptor values with embedded definitions of smallest cosine distance to the summary are returned to form a "short list". Lastly, the t short-listed values and their definitions are provided to a GPT model, along with an ICL prompt similar to the other descriptors', so that the GPT model will return its top prediction for the descriptor's value.

Having predicted a value for each of the descriptors, Data PAL has constructed the profile of an ideal dataset for retrieval. However, a dataset with this specific combination of descriptors may not exist in the data project. We filter all available datasets by each descriptor's prediction sequentially, skipping descriptors that cannot be satisfied due to previous descriptor values. Afterwards, we select a best-match dataset for retrieval.

2.3 Analysis Component

The Analysis Component instantiates an OpenAI "Assistant" GPT model with the proprietary Code Interpreter Tool [1], allowing GPT to execute code for tasks such as data visualization and mathematical computations. This GPT agent is prompted with the full conversational history and all retrieved datasets, allowing the model to generate an informed response to the conversation using Retrieval-Augmented Generation (RAG) techniques [13].

Oftentimes, scientific datasets are stored in specialized file formats that GPT cannot natively interpret. In this case, we use a specialized ICL prompt instructing GPT to install any Python libraries needed for interacting with a given data project's file formats.

3 CMIP6 Implementation of Data PAL

In order to be adaptable to a wide variety of data projects, the description of Data PAL in Section 2 is intentionally abstract. We now discuss specific details and implementation decisions for the GISS CMIP6 datasets.

3.1 Structure of GISS CMIP6

GISS CMIP6 contains the evaluation data and outputs of six climate models. Its 343,119 datasets simulate more than 400 different variables over 90,000 years of the Earth's past and future climate. CMIP6's modeling tasks are referred to as Model Intercomparison Projects (MIPs), each of which contain different sub-tasks, called Experiments. Beyond the large size and diversity of the CMIP6 datasets, we chose to implement Data PAL on CMIP6 because this data project is available online to the general public[1].

We use six descriptors for the CMIP6 datasets. These descriptors include *Variable* (the dependent variable measured), *Start* and *End Year* (the range of years covered) and *Temporal Resolution* (whether Variable is measured hourly, monthly or yearly between Start and End Year), along with the *MIP* and *Experiment* to which the dataset belongs.

3.2 Retrieval Component

We consider Variable to be a specialized descriptor, while prior interactions with GPT-4 demonstrated that the model possesses an understanding of the concepts behind MIP, Experiment, Start and End Year, and Temporal Resolution. The GPT prompts used to predict each descriptor are listed in Appendix A. As detailed in Section 2, each non-specialized descriptor is predicted by prompting GPT with the conversational summary, plus information such as which descriptor to predict and the descriptor's set of possible values.

Variable, however, is more challenging: there are 419 unique Variables in GISS CMIP6, each with a precise, technical definition. We use the three-step specialized descriptor technique introduced in Section 2.2 with $t = 10$, relying on definitions of each Variable that are publicly available online[2]. In Section SEC, we compare the performance for predicting Variable using the three-step technique versus the simpler process used for the non-specialized descriptors to demonstrate that the three-step process significantly improves prediction accuracy.

3.3 Analysis Component

CMIP6 datasets are stored in the specialized geospatial NetCDF format [19], which GPT cannot natively interpret. We use a specialized ICL prompt instructing GPT to install xarray [10], the Python library for interacting with NetCDF files.

Because the Analysis Component is tasked with formulating the user-facing responces, this component interacts with an accompanying Graphical User Interface (GUI) as well: we pair Data PAL with a custom UI using the Python Streamlit library [12], which allows developers to quickly create and share custom web apps. The UI mimics a conversational text message format, with user input on the right side and the LLM response on the left. The history of the conversation is displayed, along with any plots generated by the Analysis Component.

4 Evaluation

We now evaluate the performance of Data PAL for GISS CMIP6. Because this is the first work that attempts to create a unified retrieval/analysis framework for general large data projects, it is necessary to define a benchmark for this task. To do so, we crowdsource our own retrieval/analysis evaluation dataset, which is described in Section 4.1. Then, we present and justify our experimental setup, metrics and baselines in Section 4.2.

4.1 Evaluation Dataset

We crowdsource a set of 35 GISS CMIP6-related queries from NASA scientists. We provide a small sample of our crowdsourced evaluation dataset in Table 1. Each query is a standalone question to be answered by Data PAL, rather than a multi-turn conversation that must first be summarized. These single-turn queries are easier to crowdsource from volunteers than multi-turn conversations (despite

[1] https://github.com/PCMDI/cmip6-cmor-tables/tree/main

[2] https://portal.nccs.nasa.gov/datashare/giss_cmip6/

Table 1: A subset of the evaluation dataset queries.

Query	Variable	Start Year	End Year	Temporal Resolution	MIP	Experiment
Are there going to be increased heatwaves in South America under SSP370 for 2085?	tasmax	2085	2085	day	ScenarioMIP	ssp370
Show me in the future, all the suitable places that wheat could grow	clt, pr, etc	2025			ScenarioMIP	
Show me the expected average winter ice coverage for Lake Ontario is 2050?	sbIIs, sftgif	2050	2050	month	ScenarioMIP	
Plot the change in cloud cover from 1930 to 2015	clt	1930	2015	month	CMIP	historical
What are the projected changes in global ocean salinity by 2050 under SSP126?	so	2025	2050	month	ScenarioMIP	ssp126

this, we still pass each query through the Retrieval Component's conversational summarization step in order to shorten the query).

After manually annotating each of the 35 queries, we perform semantic variation to augment our evaluation dataset. For each query, we ask GPT-4o to rephrase the query five different ways for a total of set of 210 queries (35 original plus 175 augmented). As such, the manual annotations for each original crowdsourced query can be used for the semantic variation queries as well.

We use this semantic variation method to augment our dataset because of its simplicity, and also because of the quality of the generated queries as measured by cosine distance. We refer to the original, crowdsourced queries as "parents" and their rephrased queries as "children". As demonstrated by the plots of SciBERT [4] and OpenAI embedding cosine distances between each child query and its parent in Figures 3 and 4, the augmented queries have small– but non-zero– distances to their parent query. As a result, the augmented queries tend to have similar meanings to the crowdsourced queries, without being identical.

To help foster future research in this area, we are currently in the process of gaining the rights to release our evaluation dataset publicly.

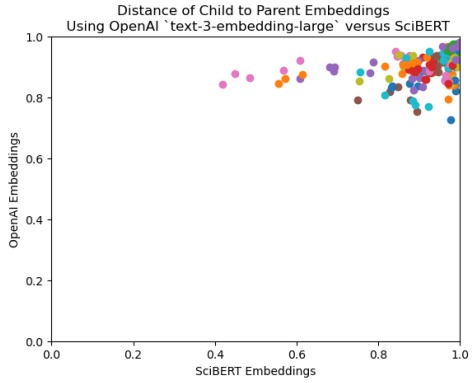

Figure 3: A comparison of child queries' embedding cosine distances to their parents' embeddings, using SciBERT [4] or OpenAI's text-3-embedding-large model. Each color corresponds to one parent query. We observe that distances produced by the OpenAI embeddings are generally closer to 1 than the SciBERT embedding distances.

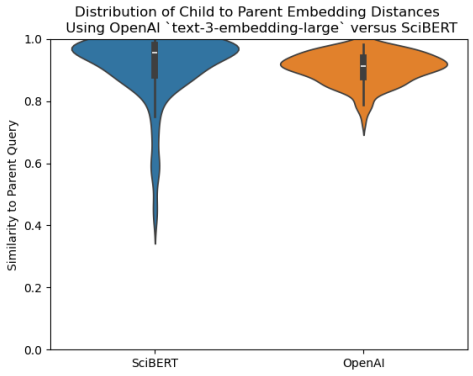

Figure 4: The distributions of cosine distances between child and parent queries, using SciBERT or OpenAI's text-3-embedding-large model. The range of distances for OpenAI (right) are closer to 1, despite a lower *average* distance than SciBERT embeddings. The SciBERT distribution exhibits a longer tail of lower-similarity embeddings as well.

Table 2: Variable prediction accuracy. Data PAL (denoted DP) outperforms both the RAG (denoted 3.5) and the embeddings-based baselines by over 25%.

	Chosen	Top 1	Top 3	Top 5	Top 10	Top 100
DP 4o	$\mathbf{65.4 \pm 1.0}$	$\mathbf{39.4 \pm 0.7}$	$\mathbf{53.7 \pm 0.5}$	$\mathbf{56.2 \pm 0.5}$	$\mathbf{68.6 \pm 0.5}$	77.1 ± 0.0
DP 3.5	62.5 ± 0.5	36.8 ± 0.5	48.9 ± 0.3	51.9 ± 0.0	67.8 ± 1.0	77.1 ± 0.0
3.5	8.6 ± 0.0					
Embed	35.7 ± 0.5					

4.2 Evaluation Setup

We evaluate Data PAL retrievals in two phases. As described in Section 2, we first predict the values of each descriptor. We begin by assessing the accuracy of these predictions in Section 4.3. Next, in Section 4.4, we discuss the accuracy of the dataset that is ultimately retrieved.

For the Temporal Resolution, MIP and Experiment descriptors, accuracy is calculated as the percentage of queries for which the descriptor agent correctly predicts the descriptor's gold label. For the Year descriptor, 50% accuracy is awarded for a correct start- *or* end-year prediction, while 100% accuracy is awarded for correct start- *and* end-year predictions.

In all experiments, we include Data PAL with both GPT-4o and GPT-3.5, which we refer to as Data PAL-4o and Data PAL-3.5. Furthermore, all experimental results are averaged across three runs for any non-deterministic approach. We focus here on the evaluation of dataset retrievals, with plans to evaluate the Analysis Component outlined in Section 4.5.

4.3 Descriptor Prediction

Due to the additional challenges and complexities of Variable prediction, we first focus on this descriptor individually before comparing the performance of the remaining descriptors.

4.3.1 Variable Prediction

Accurate prediction of Variable presents additional challenges compared to the other descriptors, because the descriptor agent must correctly choose one Variable from among the 419 unique values represented within the GISS CMIP6 datasets, as opposed to tasks such as MIP prediction, where there are only two possible values. Furthermore, many of these Variables are described in specialized scientific language, unlike descriptors including Year and Temporal Resolution, which have arguably more intuitive definitions.

Due to these complexities, along with the corresponding added complexity of our three-step Variable prediction process compared to other descriptors' prediction processes, we provide a more thorough evaluation of the Variable prediction.

Variable accuracy is calculated as the percentage of queries where the variable is correctly predicted. Several of the evaluation queries have multiple correct variable answers; in these cases, the predicted variable is correct if it equals to any of the variables within the gold label set.

Baselines: We implement two Variable prediction baselines. The first is embedding-based: we use OpenAI's largest embedding model, 'text-3-embedding-large' to create 3072-dimensional embeddings of natural-language descriptions for each of the 419 variables, as well as each evaluation query. A Variable with an embedded description of minimum cosine distance to the embedded query is chosen as this baseline's description.

Our second variable baseline represents a simple RAG pipeline: we provide GPT 3.5 with a table of all 419 Variables and their natural-language descriptions, along with an ICL prompt to return the Variable best-suited to answer the evaluation query.

Results: Data PAL-4o is 29.7% more accurate on average than the best-performing baseline. The RAG baseline using GPT-3.5 without Data PAL reaches only 8.6% accuracy. Although RAG-3.5 and Data PAL-3.5 both use the GPT-3.5 model, *the significantly lower performance of RAG-3.5 highlights*

Table 3: Non-specialized descriptor prediction accuracy. Data PAL (denoted DP) outperforms the keyword baselines on all descriptors except Experiment.

	Year	Temporal Resolution	MIP	Experiment
DP 4o	$\mathbf{94.6 \pm 0.1}$	$\mathbf{83.8 \pm 0.8}$	$\mathbf{88.6 \pm 1.3}$	57.1 ± 1.7
DP 3.5	86.0 ± 0.5	68.3 ± 0.5	57.8 ± 0.3	62.1 ± 0.5
Keywords	64.3	12.9	14.3	**63.3**

the effectiveness of Data PAL-3.5's Variable selection approach. These findings are summarized in the "Chosen Variable" column of Table 2.

The remaining columns of Table 2 highlight the importance of our three-step specialized descriptor prediction process. Each of these "Top-t" columns reveal the percent of queries for which a correct Variable prediction is contained within the top-t description embeddings of closest distance to the summary.

According to the "Top-1" column, if we were to rely only on embeddings alone by returning a Variable with a description embedding *the closest* to the summary, then Variable would only be predicted correctly for only 39.4% or 36.8% of queries, when using Data PAL-4o or Data PAL-3.5 respectively. Meanwhile, because a correct Variable is contained within the top-10 closest description embeddings on about 68% of queries, our third step of choosing the Variable prediction out of the top-10 closest using GPT is accurate for 65.4% or 62.5% of queries respectively.

4.3.2 Prediction of Other Descriptors

Baselines: We compare Data PAL's performance on the remaining descriptors against keyword-based methods. For instance, the Temporal Resolution baseline predicts 'hr', representing hourly resolution, if the query contains any of the words 'hr', 'hour' or 'hourly'. The Year baseline uses a regular expression to find all years in the query; returning the first and last matches as the start and end years respectively, or 'None' for the start and end years if no matches are found.

Comparison of other Descriptor predictors As demonstrated in Table 3, Data PAL-4o exceeds the baseline by significant margins on all descriptors except Experiment, while Data PAL-3.5 achieves the second-best performance for all descriptors. The keyword baselines especially struggle with Temporal Resolution, which takes four unique values, and MIP, which takes three unique values, performing worse than uniform random guessing.

The high performance of the Experiment keyword baseline is at first glance surprising: this baseline outperforms even the baselines tailored for the apparently simpler descriptors of Temporal Resolution and MIP, that take fewer possible values. There are 15 distinct values (including undefined) for Experiment within the GISS CMIP6 datasets, with esoteric names including '1pctCO2' and 'ssp460'. However, the Experiment descriptor plays to the keyword baseline's one strength relative to Data PAL's reliance on GPT: willingness to predict that this descriptor is undefined for a given query.

Of the 210 evaluation queries in the augmented dataset, 84 queries (40%) are labeled as undefined in the Experiment category. Only two of the 84 queries with undefined Experiment mention one of the other possible Experiment values, so the keyword baseline achieves $\frac{82}{84} \approx 97.6\%$ accuracy on this sizable subset of queries. Data PAL-3.5 and Data PAL-4o are reluctant to predict that any descriptor is undefined, achieving on average only $\frac{30}{84} \approx 30.0\%$ and $\frac{23.7}{84} \approx 28.2\%$ accuracy on these queries.

In future, swapping out the GPT-based Data PAL Experiment classifier with the keyword-based Experiment classifier would be an easy way to improve Data PAL's performance. Additionally, improvements to Data PAL's ICL prompts with additional encouragement to predict undefined descriptors, such as by including the prior for undefined descriptors (i.e., 40% for Experiment), may improve the performance of Data PAL's GPT descriptor classifers in these situations.

4.4 Results: Accuracy of Retrieved Dataset

Next, we examine Data PAL's capacity to select a dataset for retrieval by combining its individual descriptor predictions. This step poses some new challenges that are not considered in the prior step of predicting descriptor values. Namely, CMIP6 contains many inter-descriptor dependences: the choice of one descriptor limits the set of possible choices for the other descriptors. For instance, though there are 14 unique Experiment values represented in the GISS CMIP6 datasets, only two of these values occur in the datasets with a MIP value of 'CMIP'. Each of the individual descriptor classifiers works independently, without regard for these dependencies, and the final step of choosing a dataset to retrieve must rectify any mutually-exclusive values predicted in the prior step.

Similarly to the evaluation of the predicted descriptors, we grade the retrieved dataset as follows: for each query and each descriptor, the chosen dataset is considered accurate if the retrieved dataset's value in that descriptor equals the descriptor's gold label for that query.

Baselines: We implement three baselines for this task.

Two baselines are combinations of the descriptor prediction baselines introduced in Section 4.3, along with Data PAL's process of using these predictions to select the retrieved dataset. The first baseline, called E+K, uses the embedding approach to predict the Variable. The second, called 3.5+K, predicts Variable using the GPT-3.5 RAG baseline. These baselines both rely on the keyword-based approaches to predicting Start/End Year, Temporal Resolution, MIP and Experiment.

The third baseline, called 3.5, is a single-step RAG approach. We provide GPT-3.5 with a table of all 343,119 GISS CMIP6 datasets and prompt the model to choose a dataset appropriate to the query.

Results: Table 4 compares the accuracy of the dataset selection methods. *Data PAL-4o and Data PAL-3.5 are more accurate in the Variable descriptor than all baselines, by large margins.*

Start and End Year see lower performances. The 3.5 baseline achieves the highest accuracy, at 62.7%. While this baseline chooses its dataset in one step, the other methods are constrained by their choice of Variable before attempting to select their predicted Start Year. When the predicted combination of Variable and Start Year does not exist in the GISS CMIP6 datasets, Data PAL, 3.5+K and E+K opt for their predicted Variable instead of their predicted Start Year. We refer to this effect of degraded performance due to constraints from prior descriptors as *prior descriptor limitation*.

Despite its freedom from prior descriptor limitation, the 3.5 baseline struggles at selecting relevant datasets. In fact, this method sees the second-lowest accuracy for the Variable descriptor, and is *outperformed by Data PAL-3.5 or Data PAL-4o in all descriptors but Start Year*.

We see the effect of prior descriptor limitation even more clearly in Temporal Resolution. As presented in Section 4.3, the keyword baseline for predicting Temporal Resolution achieved only 12.9% accuracy (worse than random guessing), versus 83.8% and 68.3% for Data PAL-4o and Data PAL-3.5. Despite the low performance of the keyword Temporal Resolution predictions, which are used identically by both E+K and 3.5+K, we see that datasets selected by E+K and 3.5+K perform similarly to datasets selected by Data PAL on the accuracy of Temporal Resolution.

Due to the prior descriptor limitation effect, we find a need to adjust assessments of Data PAL, E+K and 3.5+K's retrieved datasets' descriptors on the basis of how limited each method is by its prior descriptor choices. The design of a clearer metric for evaluating the retrieved datasets is a priority for future research. Despite these challenges, *the competitive performance of Data PAL is demonstrated in its high Variable accuracy, along with its highest or near-highest accuracy on four of the five other descriptors*: End Year, Temporal Resolution, MIP and Experiment.

4.5 Analysis Evaluation

To evaluate the effectiveness of the Analysis Component in Data PAL, we are in the process of conducting a series of assessments, which we discuss in-turn.

Descriptor Interpretation: The first evaluation focuses on the Analysis Components's ability to correctly interpret datasets recieved from the Retrieval Component. In particular, we assess the Analysis Component's ability to identify descriptors of a dataset retrieved in response to a user query.

Table 4: Accuracy of retrieved dataset. Data PAL (denoted DP) achieves top performance on 4/6 descriptors. Baselines are, in order: keywords with embedding-based or RAG Variable prediction and a single-step RAG dataset selection approach.

	Variable	Start Year	End Year	Temporal Resolution	MIP	Experiment
DP 4o	$\mathbf{65.4 \pm 1.0}$	26.7 ± 0.5	30.6 ± 0.7	80.0 ± 0.5	$\mathbf{94.0 \pm 0.3}$	77.9 ± 0.7
DP 3.5	61.0 ± 0.5	31.9 ± 1.0	$\mathbf{38.6 \pm 0.0}$	67.0 ± 0.3	91.7 ± 0.3	$\mathbf{86.0 \pm 0.7}$
E+K	8.6 ± 0.0	0.0 ± 0.0	5.7 ± 0.0	77.1 ± 0.0	42.9 ± 0.0	40.0 ± 0.0
3.5+K	35.7 ± 0.5	16.0 ± 0.3	35.1 ± 0.3	$\mathbf{81.3 \pm 0.3}$	74.6 ± 0.3	67.9 ± 0.3
3.5	11.4 ± 0.0	$\mathbf{62.7 \pm 0.3}$	37.0 ± 0.3	32.9 ± 0.0	85.6 ± 0.3	69.5 ± 0.0

This evaluation will be performed using the same evaluations dataset introduced in Section 4. The performance of our method will be compared across GPT-4o, GPT-3.5, and the standard ChatGPT interface [1]. We will use accuracy as our principal performance metric for each descriptor, in a similar fashion to evaluation of the Retrieval Component.

Plot Generation Capability: This evaluation measures the component's capability to generate a plot when appropriate, regardless of the plot's correctness. For each crowdsourced query in our evaluations dataset, we will manually annotate the query with a 1 if the query's response should produce a plot, and with 0 otherwise. Performance will be evaluated using accuracy.

User Satisfaction: The final evaluation is a user study. We will assess user satisfaction with Data PAL's responses to a fixed set of n queries for U participants. This evaluation will involve a diverse group of users, from novices to experts, who will use the system and provide satisfaction ratings on a scale from 1 to 5. The metric for this evaluation will be the average satisfaction score, calculated as:

$$\text{Average\_Score} = \frac{1}{n} \sum_{q=1}^{n} \frac{\sum_{u=1}^{U} \text{satisfaction}(q, u)}{U},$$

where n is the total number of queries and satisfaction(q, u) is the satisfaction of the u-th user on Data PAL's response to the q-th query.

5 Related Work

Information Retrieval [20] has been an area of artificial intelligence research for decades. Recent approaches have relied on the advances of LLMs [15, 21, 8]. RAG is a related task, where the LLM accesses an external knowledge base to better-inform its outputs [13, 14]. However, many of these works are focused on retrievals from natural-language datasets, as opposed to specialized modalities such as geospatial data.

Two notable exceptions include Chen et al. [5] and Zhang et al. [22], which create RAG systems for road design and satellite control, respectively. While these works represent important advances in RAG, they are highly fitted to their specific applications. Their designs offer few insights into the design of a RAG system for climate change data, or for large scientific datasets more broadly.

LLM researchers have given much attention lately to the task of automated data analysis [16, 9], with the GPT Data Analysis tool [2] particularly relevant to these efforts. Similar to the problem of RAG, however, these works have primarily focused on domains such as natural language and pure mathematics. ICL is also a popular topic of LLM research [17, 6], allowing LLMs to perform novel tasks by following instructions in a text prompt instead of undergoing further training or finetuning.

6 Conclusion

We present Data PAL, a system for the retrieval and analysis of data from large scientific data projects using conversational English. It is our hope that Data PAL will be useful for accelerating research in scientific fields, providing greater exposure to and understanding of these fields to the non-technical public, and fostering future works that will further improve upon Data PAL.

References

[1] Openai platform. URL https://platform.openai.com.

[2] Improvements to data analysis in chatgpt. URL https://openai.com/index/improvements-to-data-analysis-in-chatgpt/.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] I. Beltagy, K. Lo, and A. Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[5] J. Chen, W. Xu, H. Cao, Z. Xu, Y. Zhang, Z. Zhang, and S. Zhang. Multimodal road network generation based on large language model. *arXiv preprint arXiv:2404.06227*, 2024.

[6] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[7] V. Eyring, S. Bony, G. A. Meehl, C. A. Senior, B. Stevens, R. J. Stouffer, and K. E. Taylor. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.

[8] J. Gavilanes, Y. Bozhilov, U. Dodeja, G. Valtas, and A. Badrajan. Use of llm for methods of information retrieval.

[9] S. Hong, Y. Lin, B. Liu, B. Wu, D. Li, J. Chen, J. Zhang, J. Wang, L. Zhang, M. Zhuge, et al. Data interpreter: An llm agent for data science. *arXiv preprint arXiv:2402.18679*, 2024.

[10] S. Hoyer and J. Hamman. xarray: Nd labeled arrays and datasets in python. *Journal of Open Research Software*, 5(1):10–10, 2017.

[11] K. Katz, O. Shutov, R. Lapoint, M. Kimelman, J. R. Brister, and C. O'Sullivan. The sequence read archive: a decade more of explosive growth. *Nucleic acids research*, 50(D1):D387–D390, 2022.

[12] M. Khorasani, M. Abdou, and J. H. Fernández. Web application development with streamlit. *Software Development*, pages 498–507, 2022.

[13] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[14] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*, 2022.

[15] Z. Liu, Y. Zhou, Y. Zhu, J. Lian, C. Li, Z. Dou, D. Lian, and J.-Y. Nie. Information retrieval meets large language models. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1586–1589, 2024.

[16] P. Ma, R. Ding, S. Wang, S. Han, and D. Zhang. Demonstration of insightpilot: An llm-empowered automated data exploration system. *arXiv preprint arXiv:2304.00477*, 2023.

[17] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[18] OpenAI. Introducing chatgpt. URL https://openai.com/index/chatgpt/.

[19] R. Rew and G. Davis. Netcdf: an interface for scientific data access. *IEEE computer graphics and applications*, 10(4):76–82, 1990.

[20] A. Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4): 35–43, 2001.

[21] C. Zhai. Large language models and future of information retrieval: Opportunities and challenges. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 481–490, 2024.

[22] R. Zhang, H. Du, Y. Liu, D. Niyato, J. Kang, Z. Xiong, A. Jamalipour, and D. I. Kim. Interactive generative ai agents for satellite networks through a mixture of experts transmission. *arXiv preprint arXiv:2404.09134*, 2024.

A Method

Table 5: The ICL prompts used by Data PAL's Retrieval Component, edited slightly for brevity.

Descriptor	ICL Prompt
Variable (1)	You are a climate scientist and expert on CMIP6. Given a colleague's query, describe what CMIP6 variable you would use to answer the query. For instance, you might want a rainfall-related variable for questions about drought. For a query about days below freezing, you might want a variable describing minimum temperature instead of average temperature. Formulate your response as a detailed list of keywords. Be specific because a lot of CMIP6 variables are very similar but there is only one correct answer to these queries.
Variable (2)	RETURN A ONE-WORD RESPONSE: You are an expert climate scientist working with the CMIP6. Following, is a colleague's climate analysis query and a list of 10 CMIP6 variables with their descriptions. From these 10 variables, choose the variable best-suited to answer the colleague's query. Return ONLY the variable's name and nothing else. For instance, return 'tas', or 'zostoga', or 'sithick' alone, no explanation, no alternative answer, nothing else.
Start/End Year	You are an expert climate scientist. Does the following CMIP6 query require or specify a year range for the data required to answer the query? If yes, provide the year range in format START-END, for instance 1960-1970 or 2100-3100. If no, respond NA-NA. If only the start or end is specified, provide just that year in format START-NA (eg 2100-NA) or NA-END (eg NA-1900). Provide only the year range in this format and nothing else.
Temporal Resolution	You are an expert climate scientist. Is the following CMIP6-related query best answered using data gathered at which of the following resolutions? A. hour B. day C. month D. not applicable, none of the above, or unclear Respond with only the one letter corresponding to your choice and nothing else. If a query does not specify any given temporal resolution, like the query "plot average temperature", then choose option D.
MIP	You are an expert climate scientist working with CMIP6. Here is a list of the MIPs you work with: CMIP, ScenarioMIP To answer the following query, which of the above experiments would you use? Return JUST the name of the experiment and nothing else. If the choice of experimentdoes not matter, return 'None'.
Experiment	You are an expert climate scientist working with CMIP6. Here is a list of the experiments you work with: To answer the following query, which of the above experiments would you use? Return JUST the name of the experiment and nothing else. If the choice of experimentdoes not matter, return 'None'.