Harnessing Large Language Models for Research Institutions: an example based on NASA / JPL use-cases

Steffen Mauceri[1], Asitang Mishra[1], Ryan M. McGranaghan[1], Ashish A. Mahabal[2], Lukas Mandrake[1], Benjamin Smith[1], Dustin J. Graf[1], Benjamin Nuerenberger[1], Alice R. Yepremyan[1], Brian Wilson[1], Amanda Towler[1], Kay Y. Pak[1], Miles B. Pellazar[1], Daniel J. Crichton[1]

1 Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
2 California Institute of Technology, Pasadena, CA, USA

Abstract:

In the era of information overload, how can research institutions effectively manage and utilize the wealth of data at their disposal? State-of-the-art Large Language Models (LLMs) could hold the key. These models have the potential to revolutionize work processes at research institutions, aiding tasks such as ideation, summarizing text, generating code, and question answering. These applications could significantly enhance scientific research and knowledge discovery in earth and space science, aid decision making, and enable better alignment of challenges with the right experts.

We conducted a survey of current thrusts and use cases from diverse user groups across the Jet Propulsion Laboratory (JPL), using a comprehensive approach that combines quantitative and qualitative research methods. Focusing on both available capabilities from industry providers and custom-built solutions, we'll share strategic recommendations for harnessing LLM capabilities and discuss their implications for governance workflows.

We also explored how LLMs can enhance knowledge management and discovery by making complex scientific information more accessible and easier to analyze. By leveraging the unique capabilities of LLMs, JPL and other research institutions can accelerate scientific discovery and technology development in earth and space science.

What are Large Language Models?

Machine Learning Models trained on huge amounts of text with a simple objective: predict the next word in a sentence or paragraph.

Examples include ChatGPT (OpenAI), Bard (Google)

What Large Language Models currently can and can not do:

Possible:
- Extract information from one or multiple documents (and provide references)
- Help with coding: Explain complex code. Autocomplete a few lines of code. Add comments to code
- Summarize / Polish: text, emails
- Brainstorming: next steps, introduction, suggest title, pros / cons, alternate views

Not Possible:
- Reason over a large catalogue of documents and return well researched answers.
- Automate research; write a complete paper
- Always provide the correct answer

Use-cases:

Using an online spreadsheet, employees across JPL self reported how they currently use ChatGPT like tools:

- Software Coding (explain code, write tests, translate languages, identify gaps, find best practice violations)
- Analysis (logs, text, extract insights)
- Draw/Design (schemadraw, programatic drawing of figures, artists renditions)
- Brainstorming (options, intros, flow, requirements, vendors, targets)
- Refine/Meta (polishing, prompting)
- Summarize (text, tables, videos, video call transcripts, emails, action items)

Addionally we conducted 1:1 interviews with selected user groups at JPL to understand other possible impactful Large Language Model use-cases and needs.

A common request was to provide more education about LLMs: understand possibilities, shortfalls, what's next, prevent misuse

Hardware / Software Engineer:
- Error analysis (lessons learned) from problem/error reports for hardware engineering
- System engineering, workflow, testing procedures
- Coding assistant: help with syntax, autocomplete code, explain unfamilar code, add unit tests, find vulnerabilities, best practice violations

Education office:
- Overwhelmed with number of applicants for summer internship opportunities. Need help matching opportutnies to candidates
- Get a lot of questions about how to set up internship opportunites, other process details from within JPL. Provide a chatbot that leads prospective mentors through the process and answers most questions.

Scientist / Data Scientist:
- Answer basic science questions.
- Gap analysis of reports like decadal survey
- Find and digest relevant papers in context
- Polish raw text
- Write / understand / document code.

Management:
- Summarize, categorize email
- Provide a knowledge source for decision making. Needs access to relevant information across the institution (e.g. Q&A over data bases)
- Finding experts in certain fields to build teams, ask for advice, ...
- Quick learning about new topics adaptable to current level of knowledge to make informed decision

Formulation:
- Propose future mission based on opportunities, science needs, constraints, newly developed technology, JPL goals
- Help with costing future missions (understand constraints, decisions, based on past missions): need framework for discussion not $$
- Knowledge transfer between employees and domains (science to engineers and engineering to scientists)

Solutions:

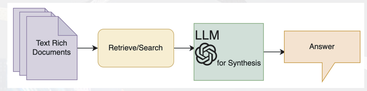
What follows is a list of currently available and future solutions to address the extracted LLM needs:

Coding Assistant: Aamazon CodeWhisperer, GitHub Copilot: A Simple assistant that everyone who codes should use. Write code faster, spend less time looking up syntax. Leads to well commented code.

Research Assistant: Many Companies popping up with big promises, but most are in early beta: e.g. Elicit (https://elicit.org/), ResearchAIde (https://www.researchaide.org/), SemanticScholar (https://www.semanticscholar.org).

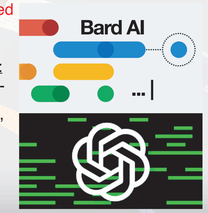
What works / is missing:
- Gain understanding of general scientific concepts / methods. Improve scientific writing
- Perform analysis over large corpus of documents to understand state of the art, other possible approaches, pros and cons for an argument
- Answers are always trustworthy

Search for Information:
Find information across unstructured data (Pdf, Word, Ppt - text rich documents etc.). Can be accomplished with Retrieval Augmented Generation based pipelines.



What works / is missing:
- Answers grounded in your data (little hallucination).
- Can provide Sources.
- Fast: can search over millions of documents in seconds.
- No training/fine-tuning needed.
- Great for finding factual answers present in your data, but Not recommended for aggregative question answering!

Discovering People Based on Expertise, Project Teaming, and Connecting People to Opportunities:
What JPL already has:
Various internal semantic technologies.
However, tools are challenging to use / query by average user (LLMs could provide more userfriendly interface). Unstructered data currently not integrated (could be done with LLMs)

Concept of an Institutional Knowledge Graph
Courtesy: IKG team

General Question Answering: ChatGPT, Bard AI:
The corpus in connection with language itself is excellent. Typical failures are when few sources are available, and where "understanding" of "concepts" rather than language is required (e.g. math, and anagram making)