

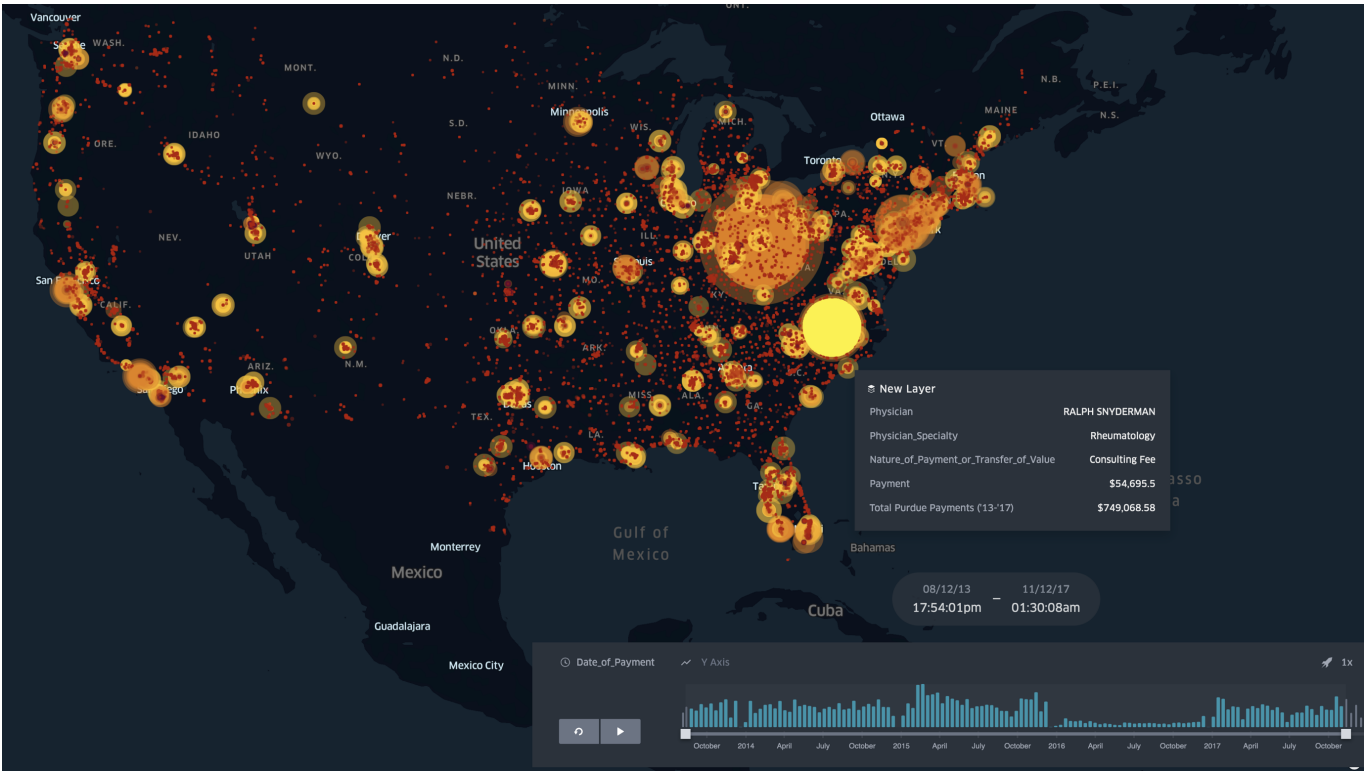
LICENSE

MIT

 LINKEDIN

► Table of Contents

A Local LLM Research Assistant



Imagine being able to rescue your research assistant from manually classifying 24 thousand entries by hand. Better yet, imagine being able to conduct research that you're passionate about even when you have limited time and no research assistants at all!

Ollama on your laptop is a great option if you have a limited budget, concerns about putting your data in the cloud, or just want to tinker with large language models (LLMs) without any platform lock-in. Mistral open-sourced their excellent 7 billion parameter model under an Apache license. The pair make a powerful combination.

This repo processes author affiliations and returns the nation & city, states

Data: PubMed metadata on JAMA articles published between 2013 and 2023 downloaded as a .nbib citation manager file
Tools: Python, Mistral 7b, AWS Bedrock, Pydantic, Pandas

Results: There were 24,801 unique first-listed-affiliations in the dataset. The LLM was able to generate responses that passed the validation criteria for 24,091 of them. The LLM was unable to produce validatable results for 709 first-listed affiliations (2.85% of the sample). Within a randomly selected 1,000-entry evaluation set that was manually checked there were 4 errors (0.4%). It's noteworthy that there were also 220 entries in which the LLM got the correct city/state/nation even though they could not be obtained from the text in the affiliation (e.g. "department of nephrology, geisinger, danville" -> Danville, PA, USA)

Built With

- Mistral.AI

- [AWS Bedrock](#)
- [AWS Cloud9](#)

[\(back to top\)](#)

Getting Started

To get started:

- Set up an AWS account [AWS](#), and log in.
- Best practice: Create a non-root user and assign that user a role with Bedrock and EC2 access permissions.
- In AWS Bedrock: manage model access -> get access to Mistral 7b
- Start a Cloud9 virtual IDE with t2.micro and as long a run window as you think you'll need (I set 1 day for this run)
- Clone the code into the Cloud9 env
- pip install boto3, pandas, and pydantic in the Cloud9 terminal window
- Eliminate the temporary security token for the Cloud9 env: Settings -> AWS Settings -> toggle temporary security token off (Note: now you'll be relying on the Bedrock permissions assigned to your IAM user role. I had a lot of issues trying to apply the Bedrock function to a pandas dataframe, though running it as a loop worked fine.)
- Hit run and watch the results

Prerequisites

This is an example of how to list things you need to use the software and how to install them.

- AWS Account
- Bedrock & Cloud9 setup w/ Mistral 7b access
- Less than \$2 in Bedrock cost

Usage

This is a quick demo of using Mistral 7b to extract geographic information from medical journal authors listed institutional affiliations. For example: Input: "department of medicine, university of pennsylvania." Response: "philadelphia, pa"

The dataset comes from a .nbib citation manager file drawn from a PubMed search for all articles published in JAMA between 2013 and 2023.

Contributing

Contributions are what make the open source community such an amazing place to learn, inspire, and create. Any contributions you make are **greatly appreciated**.

If you have a suggestion that would make this better, please fork the repo and create a pull request. You can also simply open an issue with the tag "enhancement". Don't forget to give the project a star! Thanks again!

1. Fork the Project
2. Create your Feature Branch (`git checkout -b feature/AmazingFeature`)

3. Commit your Changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the Branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

[\(back to top\)](#)

License

Distributed under the MIT License. See `LICENSE.txt` for more information.

[\(back to top\)](#)

Contact

Your Name - alexrich@duck.com

Project Link: https://github.com/alexF3/LLM_extract_geography

[\(back to top\)](#)

Acknowledgments

Great resources and people whose work make this all possible:

- [Mistral.AI](#)

[\(back to top\)](#)