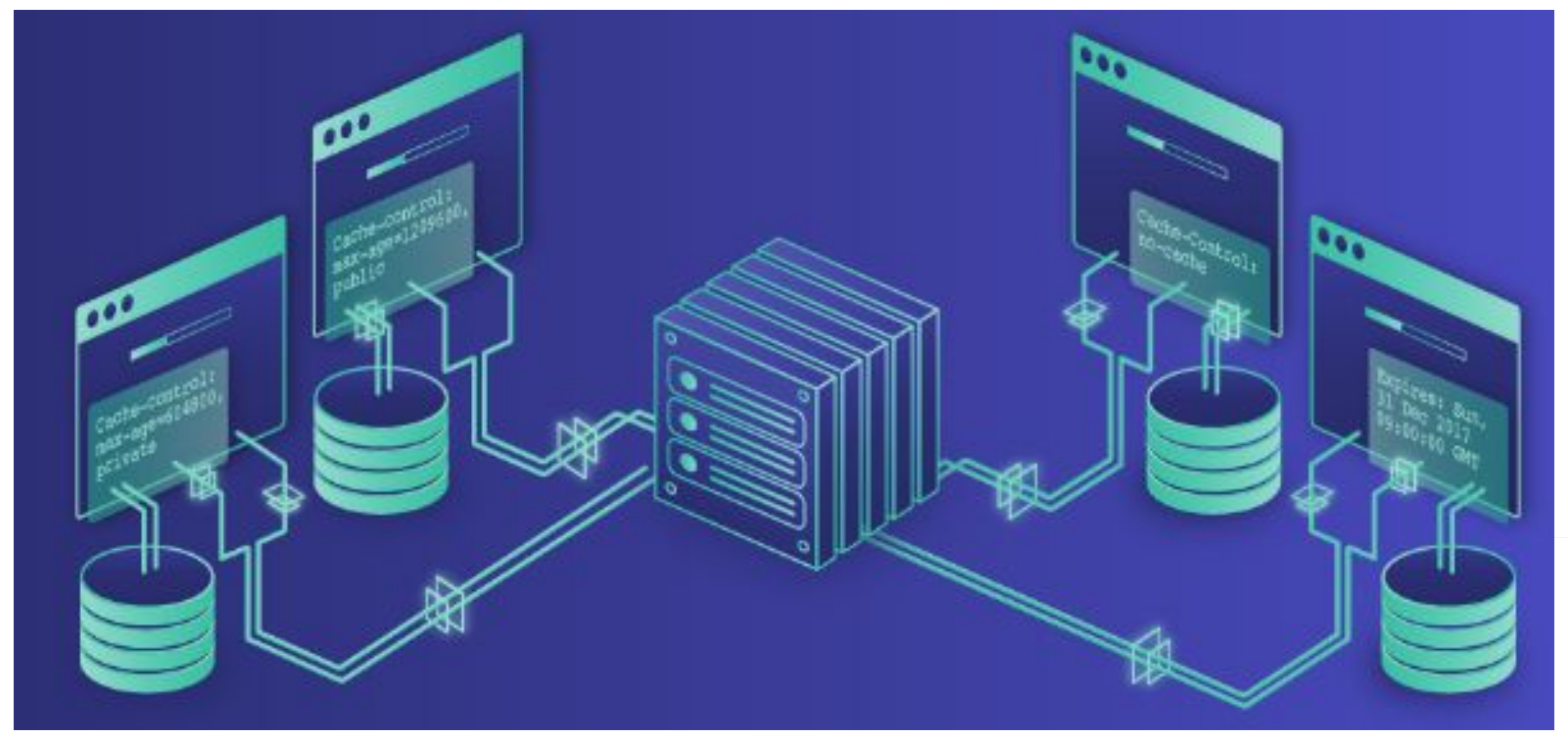
CAPSTONE DESIGN EXPO | Computer Science | CS 25-331

AI-based browser extension for knowledge management using retrieval-augmented generation and large language models

Team members: Samual Hodges, Katherine Parkyn, Alan Velasquez, Noor Tabanjeh | **Faculty adviser:** Tom Arodz, Ph.D. | **Sponsor:** VCU College of Engineering | **Mentor:** Tom Arodz

Problem Overview

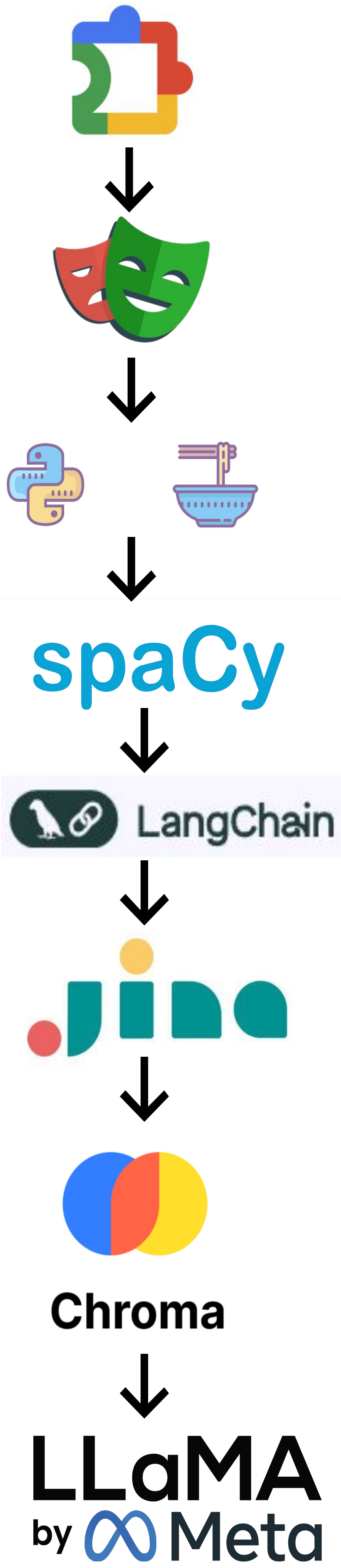
- Professionals, researchers, and students deal with significant amounts of digital information every day. More efficient solutions are needed to find what they're looking for in these data-heavy environments.



- This problem is common in fields like research, education, law, and technology, where finding the right information is essential. As digital information grows, managing and retrieving knowledge becomes a daily struggle.
- Inefficient searches lower productivity, cause stress, and add costs, with risks of burnout and environmental impact.
- This project uses AI and new technology to make organizing and locating information better. It uses advanced language models to provide smart, personalized results that are more helpful than regular search engines or manual data organization.

VCU College of Engineering

Problem Solution



User Action:
- User activates browser extension

Web Scraping:
- Playwright: Scrapes content from active web page.

Text preprocessing:
- Beautiful Soup: cleans scraped content, removing HTML tags and unnecessary whitespace.

Text Splitting:
- spaCy: Splits preprocessed text into sentences

Document Chunking:
- LangChain: Groups sentences into chunks of 10 sentences each, forming structured documents.

Embedding Creation:
- arkohut/jina-embeddings-v3: Generates embeddings for each document chunk.

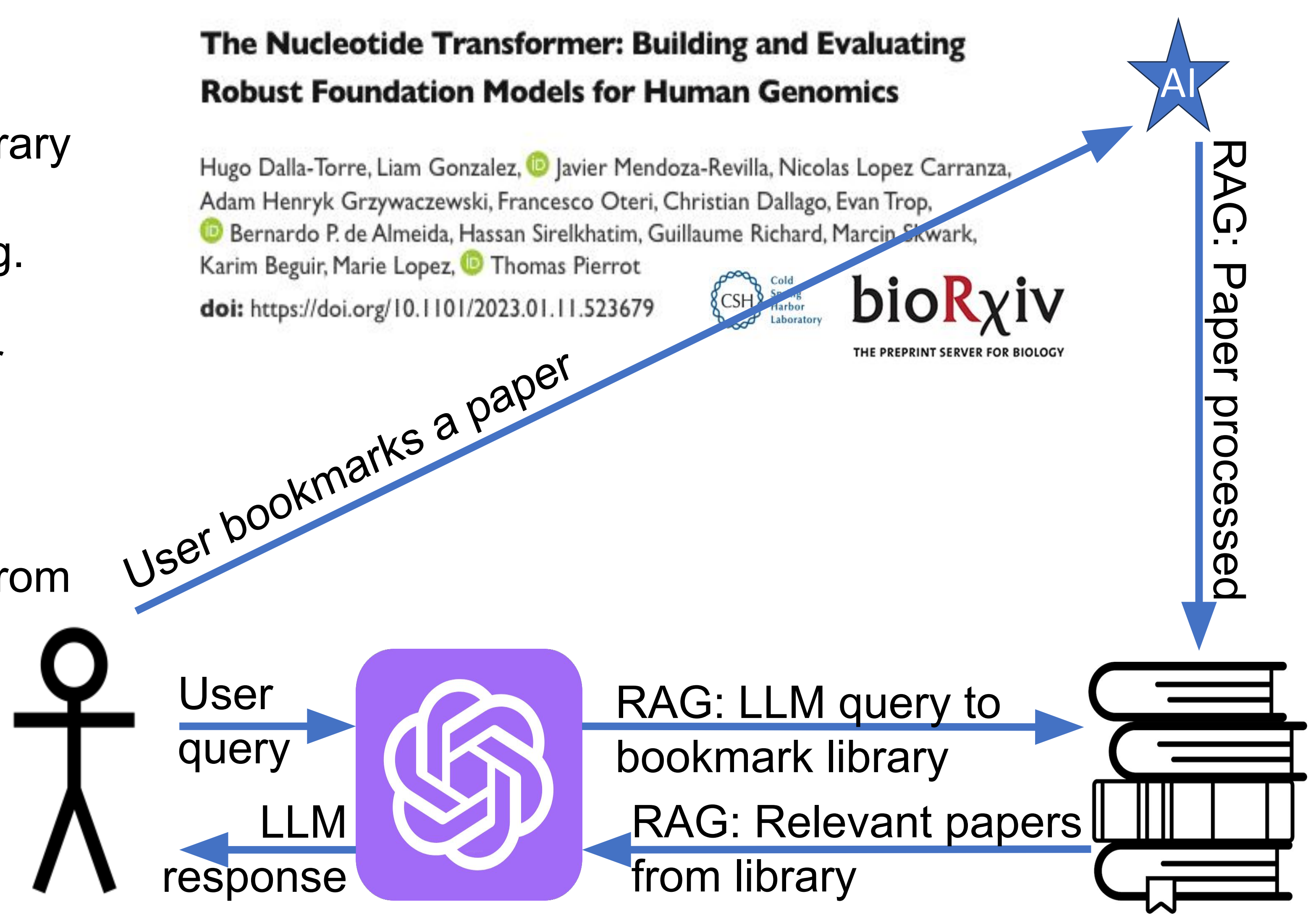
Storage in Vector Store:
- Chroma: Stores the embeddings in a vector store for efficient retrieval

Retrieval and Response:
- RAG Integration: The extension queries the vector store when a user initiates a search.
- Large Language Model (e.g., Llama): Generates a response by retrieving relevant information based on user input.

Use Case: Answering queries about "bookmarked" papers

- Bookmarked papers are encoded and stored in a database: a personal library
- When asked a query, e.g. *"In the article about Transformer systems for genomics, what was the context length?"*

the LLM uses knowledge from the personal library while formulating the answer



The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P. de Almeida, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, Thomas Pierrot

doi: https://doi.org/10.1101/2023.01.11.523679

Cold Spring Harbor Laboratory
bioRχiv
THE PREPRINT SERVER FOR BIOLOGY
AI
User bookmarks a paper
RAG: Paper processed
User query
LLM response
RAG: LLM query to bookmark library
RAG: Relevant papers from library

Future Work

- Improve response accuracy by refining language understanding.
- Implement smarter personalization so it learns user preferences.
- Add support for organizing more types of saved content, like notes or research papers.
- Enhance search accuracy to make finding old bookmarks and searches easier.

Limitations

- Speed of generating embeddings and querying AI is limited by the server it's running on.
- Browsers limit the amount of data that can be stored locally which might be an issue if we're saving embeddings.