PK-LLM : Large Language Model (LLM) for Pharmacokinetic (PK) Data Curation

Prerna Parakkat[2,3] & Matthias König[1]
[1]Humboldt-University of Berlin, Institute for Theoretical Biology, Berlin, Germany
[2]Humboldt Internship Program, [3]Vellore Institute of Technology, Chennai, India
https://livermetabolism.com, prernaparakkat@gmail.com, koenigmx@hu-berlin.de

Background

Our group has developed PK-DB [1], an open pharmacokinetics database from clinical and preclinical research. The aim of this Humboldt Internship project is to use Large Language Models (LLM) to support effective data curation for PK-DB from scientific pharmacokinetic literature.

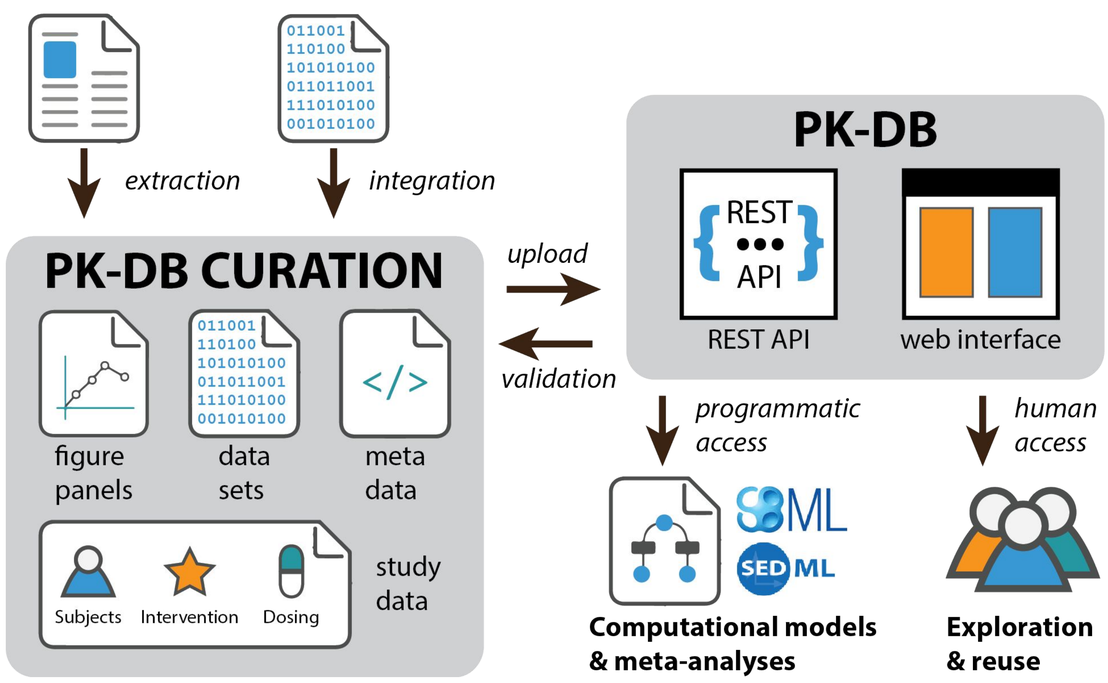

Figure 1. PK-DB overview. Schematic overview of the PK-DB data curation process. Data are either extracted from the literature (digitization of figures and tables) or imported directly (from collaborators). Figure panels, datasets, metadata and study information on subjects, interventions and dosing are curated from the publications. The uploaded study information can be accessed either programmatically via the REST API or via the web front-end. PK-LLM supports the process of data extraction from the literature and the transfer of information from PK-DB.

Methods

Retrieval Augmented Generation (RAG) using prompt engineering and data from PK-DB is used to implement an expert chat bot for the support with pharmacokinetics data curation (see Fig. 2). The vector database uses information from the full text articles curated in PK-DB. Text was extracted and cleaned as Markdown.

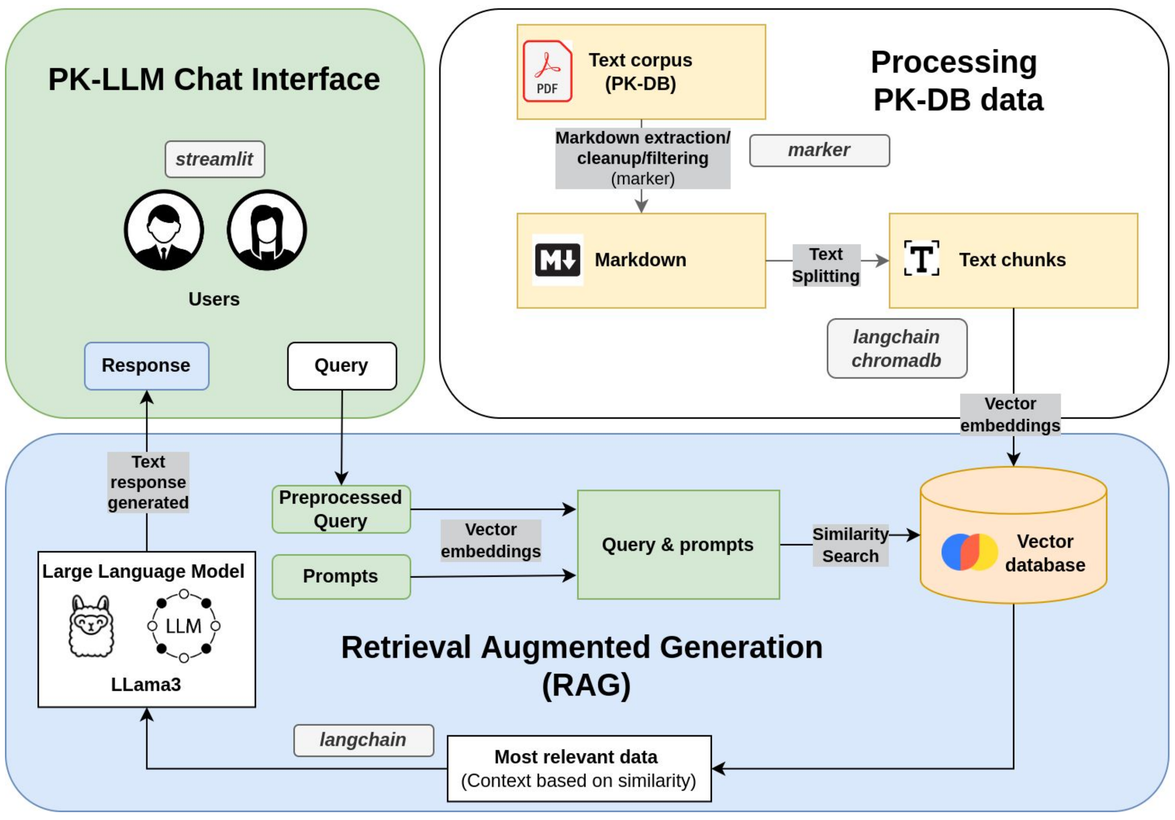

Figure 2. PK-LLM overview. Schematic overview of the PK-LLM RAG model. The PDFs (pharmacokinetic literature) in PK-DB are converted into markdown files using Marker. The markdown files are then processed, split into chunks and converted into vector embeddings using Llama3 and stored in a vector database. When a user queries the system, the given query is converted into corresponding vector embeddings. These query embeddings are then used to retrieve the most similar results from the vector database. The LLM is asked to behave as a pharmacokinetics expert system.

Results

The PK-LLM interface facilitates querying pharmacokinetics-specific information by employing similarity search to retrieve the most pertinent data from a corpus. It offers contextual information, including the source document and associated text relevant to the user query. The number of retrieved results is controlled by a set threshold parameter, ensuring only documents exceeding the threshold are selected.

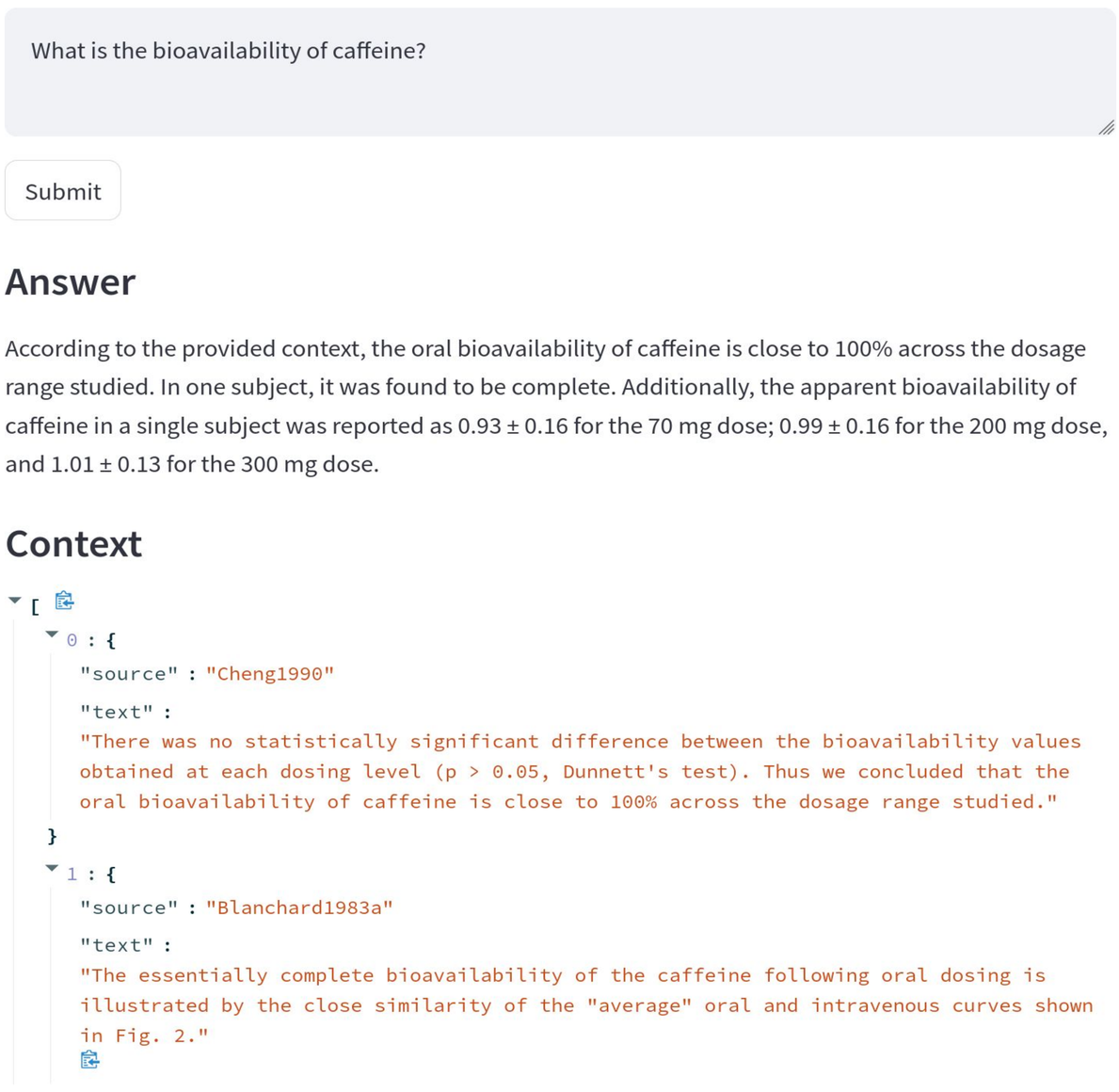

Figure 3. Example responses from PK-LLM based on user query. The most relevant context documents are retrieved from the vector database. Based on the context and prompts, the question is answered by the LLM. In the example, the bioavailability of caffeine is being investigated.

Humboldt Internship Program (HIP) Experience

The internship programme has been an exceptional experience, offering the opportunity to contribute to a fascinating project with immense potential. I have had the privilege of interacting with a diverse group of people, learning from their different perspectives and immersing myself in the vibrant life of Berlin. Navigating unfamiliar places and meeting new faces every day broadened my understanding of the world and its people. This internship has significantly deepened my knowledge of research and science. The weekly events and opportunities to explore new places with my peers made the experience even more enriching and enjoyable.

References

[1] PK-DB: pharmacokinetics database for individualized and stratified computational modeling. Grzegorzewski et al., Nucleic Acids Res. 2021, 10.1093/nar/gkaa990.

Funding

PP was supported by the Humboldt Internship Program 2024. MK was supported by the BMBF within ATLAS by grant number 031L0304B and by the DFG within QuaLiPerF by grant number 436883643 and by grant number 465194077 (SPP2311, SimLivA). This work was supported by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A537B, 031A533A, 031A538A, 031A533B, 031A535A, 031A537C, 031A534A, 031A532B).