ARA - AI-powered Research Assistant

Kaustubh Warade, Aditya Deshmukh, Devansh Parapalli, Yashasvi Thool

May 20, 2024

Resources

Software

The key software components used in this project include:

- Node.js for Development Server
- Svelte and SvelteKit as Frontend Framework
- Supabase for Backend Data Storage
- Langchain.js and Langchain.py for Language Models
- Python packages (NumPy, Pandas, Scikit-learn, etc.) for KB, and Deep Learning
- FastAPI for Orchestration Server
- Docker, Kubernetes for Deployment

Hardware

The hardware requirements for the deployment server include:

- Intel Core i7 or AMD Ryzen 5000 processor
- Nvidia RTX 4080 or better GPU
- 64 GB DDR4 RAM
- 4 TB SSD storage
- 2.5 Gbps NIC

Theory Explanation

ARA is an innovative application leveraging advanced AI technologies to revolutionize the research process. It utilizes large language models, semantic web technologies, and knowledge graphs to create an interconnected web of research information that can be processed by AI models. ARA facilitates intelligent information retrieval, contextual synthesis, effective organization frameworks, and cross-disciplinary connection identification. By automating routine tasks, accelerating data analysis, and providing quick access to relevant information, ARA aims to enhance research productivity and drive innovation across various disciplines.

Advantages

- Streamlines research workflows and enhances productivity
- Unveils hidden insights and facilitates cross-disciplinary connections
- Automates routine tasks and accelerates data analysis
- Provides quick access to relevant information
- Fosters collaboration and knowledge sharing
- Continuously learns and adapts to evolving research needs

Disadvantages

- Requires significant computational resources for deployment
- Potential for biased or inconsistent output from AI models
- Limitations in logical reasoning and common-sense understanding
- Ethical concerns related to the use of AI in research
- Potential for over-reliance on AI, limiting human critical thinking

Our Own Observations

Throughout the development and testing of ARA, we made several interesting observations:

- Integrating large language models into a user-friendly application proved challenging, as these models require significant computational resources and careful optimization to deliver real-time performance.
- Curating and cleaning training data for the AI models was a time-consuming process, as the quality and diversity of the training data heavily influenced the model's ability to generalize and provide accurate results.
- Striking the right balance between providing enough context and avoiding information overload was a delicate task when presenting research insights and connections to users. Initial models had a context window of 4096 tokens. It was later extended to 32768 tokens using a sliding window attention mechanism.
- Ensuring the reliability and transparency of AI-generated outputs was crucial, as users needed to understand the sources and potential biases or limitations of the information presented. A mechanism of citations was created, which allowed the LLM to link tokens generated to the source documents.
- Incorporating user feedback and adapting the AI models to evolving research needs required continuous monitoring and iterative improvements, making the development process dynamic and ongoing.
- The interdisciplinary nature of ARA necessitated collaboration among researchers from various domains, fostering a cross-pollination of ideas and approaches that enriched the project's outcomes.

Results and Conclusions

The development and deployment of ARA have yielded promising results, demonstrating significant improvements in information retrieval, synthesis, and organization. Researchers have reported substantial productivity gains, enabling them to focus on higher-level cognitive tasks and driving innovation. ARA's ability to continuously learn and adapt has further solidified its potential for long-term impact in the rapidly evolving research landscape.