Problem Statement

Leveraging Generative AI for Custom Chabot Development Using Large Language Models

Unique Idea Brief (Solution)

Developing a lightweight, modular Chabot framework designed specifically for efficient CPU-only deployment. The framework, dubbed AI leverages a combination of optimized LLM inference techniques, modular architecture, and domain-specific fine-tuning strategies to create a highly adaptable and resource-efficient Chabot solution.

Features Offered

1. Optimized LLM Inference:

* Implement quantization techniques to reduce the model size and improve inference speed on CPUs.
* Use mixed precision and pruning methods to further enhance performance without sacrificing too much accuracy.

2. Modular Architecture:

* Design the Chabot framework with a modular architecture, allowing easy integration of various components
* Each module can be independently optimized and updated, providing flexibility and ease of maintenance.

3. Domain-Specific Fine-Tuning:

* Develop a streamlined fine-tuning process that can be executed on CPUs, utilizing smaller, task-specific datasets to adapt the LLM to specific domains
* Incorporate transfer learning techniques to leverage pre-trained models and reduce the computational burden of fine-tuning.

4. Resource Management:

* Implement intelligent resource management to dynamically allocate CPU resources based on the Chabot’s workload and user interaction patterns.
* Utilize asynchronous processing and efficient batching strategies to maximize throughput and minimize latency.

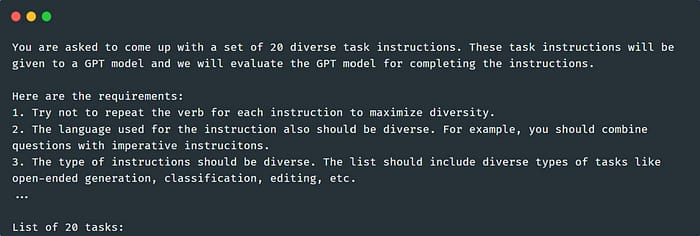
5. User-Friendly Interface:

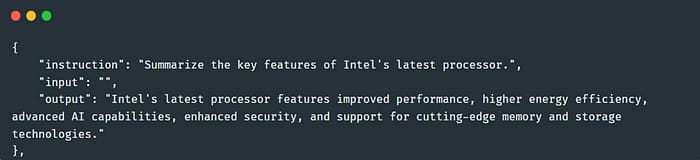
* Provide a user-friendly interface for configuring and deploying the Chabot, with tools for monitoring performance and managing conversations.
* Include pre-built templates and scripts for common Chabot applications, enabling quick setup and customization.

6. Open Source and Community-Driven:

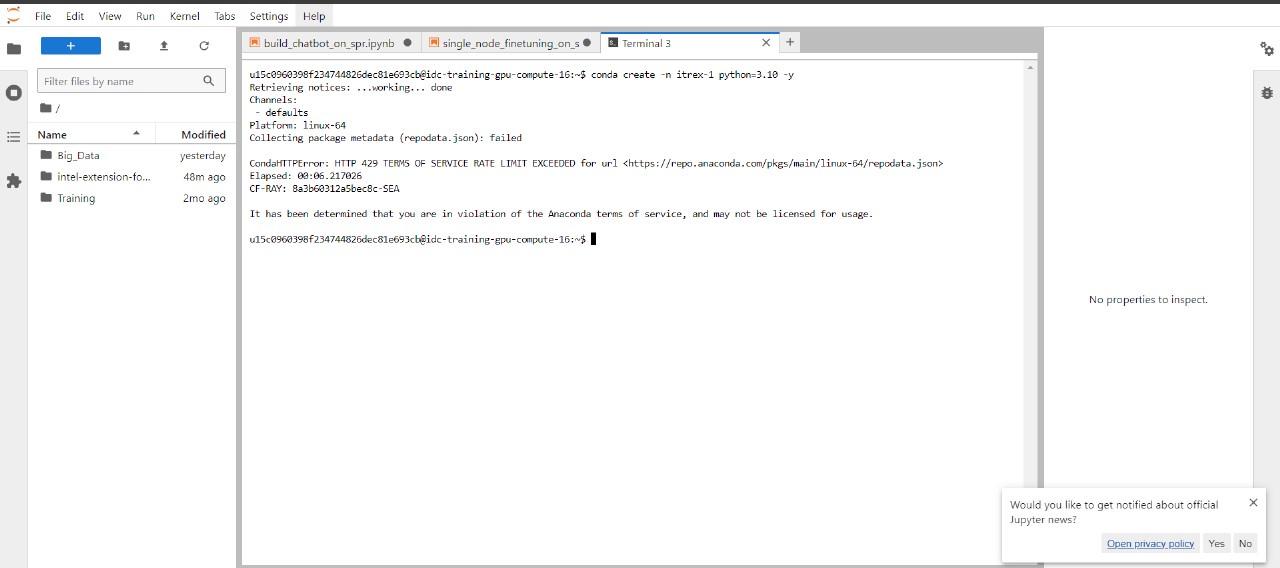
* Release the AI framework as an open-source project, encouraging contributions and improvements from the AI community.
* Create a supportive community around the framework, offering tutorials, documentation, and forums for users to share experiences and best practices.

Process flow

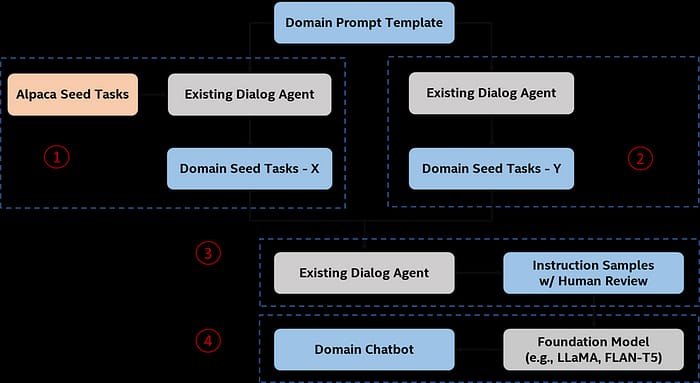








Architecture Diagram



Technologies used

1. Intel developer cloud console- Jupyter Lab
2. Google Colab
3. Intel extension for transformers
4. NeuralChat notebooks used :

1.build\_chatbot\_on\_spr.ipynb

2.single\_node\_finetuning\_on\_spr.ipynb

5. Alpaca Dataset from Standford University

6. Model: Llama 2

Team members:

KUSHAGRA GAJJAR

DHRUV HADA

SHARANYA KATNA

VAISHAK SHETTY

Conclusion

The AI framework provides an innovative solution for developing custom chatbots using Generative AI models on CPU-only environments. By focusing on lightweight, modular design and efficient resource management, this framework democratizes access to advanced AI technologies, enabling smaller organizations and individual developers to create sophisticated chatbots without the need for expensive hardware.

By addressing the challenges associated with high computational requirements and providing a practical, CPU-efficient solution, the AI framework significantly broadens the accessibility of Generative AI technologies. This framework empowers a wider range of users to leverage AI for various applications, from customer support to education, driving innovation and enhancing user experiences across different domains.

In summary, AI bridges the gap between advanced AI capabilities and resource-constrained environments, paving the way for more inclusive and widespread adoption of AI-driven Chabot solutions.