Project TLDR: Standalone Desktop application for Question-Answering and Summary using resource efficient LLMs

Project Information

Project Type: Capstone project
Student: Manu Hegde
Committee:
Prof Erika Parsons (chair)
Prof Michael Stiber
Prof Shane Steinert-Threlkeld

Overview

This project aims to develop a standalone desktop application that enables ChatGPT-like Question-Answering and Summarization on top of a corpus of documents on a user's device. The application shall embody a resource efficient implementation of a chosen Large Language Model (LLM), targeting Apple's M1/M2 hardware platform. The primary intended user base for this application is students and researchers in academia.

The motivation behind this project stems from the growing need for tools that can efficiently process and interpret large volumes of academic and research material. Current solutions often require significant manual interventions or involve sharing data with third-party servers, raising concerns about privacy and data security. By creating a resource-efficient, standalone application, this project aims to provide students and researchers with a tool that offers convenience, confidentiality, and enhanced productivity.

The project considers recent advancements in natural language processing (NLP), particularly the use of large language models (LLMs) for tasks like summarization and question-answering. The application will utilize techniques like weight quantization and low-rank adaptation to optimize LLM performance on Apple's M1/M2 architecture, including the use of Apple Neural Engine (ANE) for hardware acceleration. The project will incorporate retrieval-augmented generation to generate contextually relevant outputs using user provided corpus of text only, ensuring credibility of the information.

The expected contributions of this project include the development of a desktop application with a graphical user interface, capable of processing and summarizing large text corpora locally, without requiring an internet connection. The application will also explore the potential for running complex NLP models in resource-constrained environments, offering insights into optimizing LLMs for specific hardware platforms. Ultimately, this project aims to provide a valuable tool for researchers and students, enhancing their ability to interact with and understand extensive collections of academic materials.

Goals

The project's main goal is to develop a standalone desktop application that enables Question and Answering, Summarization on top of a repository of documents. Additionally, the resource usage of this application should be limited and predictable to allow for comfortable multi-tasking on the user's device.

Project Plan

Autumn 2024	
Week 1-3	• Shortlist 10 open-source models whose code and weights are available • Evaluate and pick one model based on resource usage and output quality tradeoffs • Finalize the model to implement for the Apple M1/M2 architecture
Week 3-5	• Evaluate model size reduction techniques applicable for the chosen model and evaluate their impact on the LLM's implementation • Finalize the implementational design of the LLM considering the exact APIs/modules of the Apple Neural Engine and CoreML API
Week 5-7	• Finalize the design of the overall TLDR desktop application • Quantize or obtain quantized FP16 and Int8 weights for the chosen model and benchmark the performance using existing open-source implementation
Week 7-10	Implement the chosen Language Model for inference, using Apple CoreML API to take advantage of ANE (Apple Neural Engine). Compare the performance with the earlier benchmarks.
Week 10-12	Implement Retrieval Augmented Generation workflow, optimizing for ANE as well as M1 CPU
Winter 2025	
Week 1-2	• Chose the appropriate framework for implementing the graphical user interface for the desktop application • Design and plan the desktop app UI components and backend modules
Week 2-4	Build wrappers around the implemented LLM and the RAG workflow to be used for integrating it with the GUI
Week 4-6	Implement the initial skeleton for the desktop GUI and integrate with the RAG workflow
Week 6-9	Implement GUI components in the desktop app for basic LLM Q&A workflow
Week 9-10	• Build features to integrate with services such as Apple notes, Zotero and Google Drive to fetch and process documents in multimodal fashion • Obtain and prepare dataset for testing. (Dataset of technical papers on arXiv.org[13] will be used as the source corpus.)
Week 10-12	Start testing and bug fixing to ensure robustness and usability of the application. • Unit testing of code for LLM and UI • Integration testing for RAG via bash scripts • Smoke testing of overall application functionality via manual interactions
Spring 2025	
Week 1-3	Continue testing: Perform experiments to obtain the pre-prompting required for simplified rephrasing of complex topics
Week 3-5	Collect user feedback and perform supervised finetuning, on the original unquantized model weights and re-quantize
Week 5-7	Collate results and prepare for presentation and defense
Week 7-9	• Prepare and submit final draft of writeup to committee • Prepare slides for defense presentation • Final project defense