SOLA - Smart Offline-First LLM Assistant

Final Project Presentation

FSCS
Flor Sanders (fps2116)
Charan Santhirasegaran (cs4347)

EECS E6692 Deep Learning on the Edge, 2024 Spring

Outline

1. Introduction
2. Proposed Solution
3. Demo
4. System Architecture
5. Implementation
6. LLM Fine-Tuning
7. Future Work
8. Conclusion
9. References

Introduction

AI Voice Assistants

- Natural user interface
- Nowadays ubiquitous

Limitations

- Not really intelligent
- Only work online
- Security and privacy concerns

Rise of LLMs

→ New wave of fancy gadgets

Voice Assistants [Apple, Google, Amazon, Samsung]
JARVIS [Wikipedia]

Proposed Solution

Privacy Concerns → Local-first Approach

- Prioritize privacy
- Keep data and compute local
- Minimize Latency

Intelligence Issue → Integrate LLMs

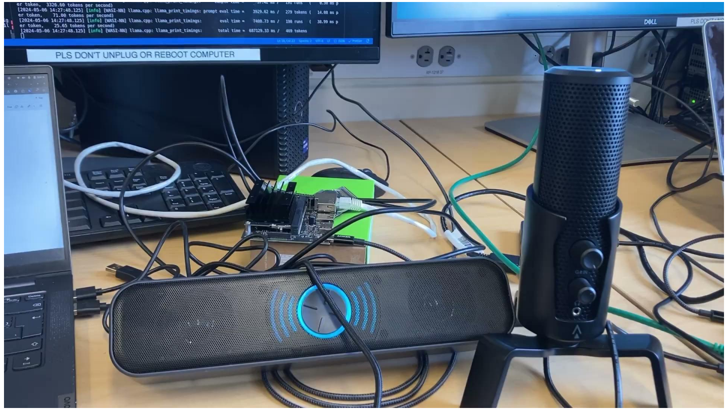
- Powerful understanding of human language
- Integrate with tools for:
  - Context-awareness
  - Internet access when needed

Jetson Nano [Nvidia]

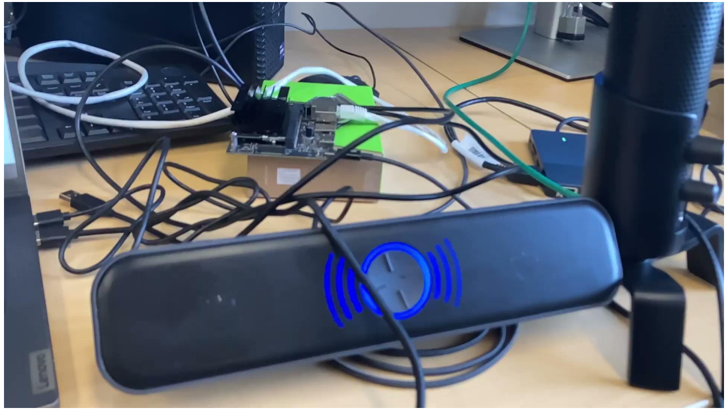
TinyLlama [Zhang et al.]

Demo - Fully Local

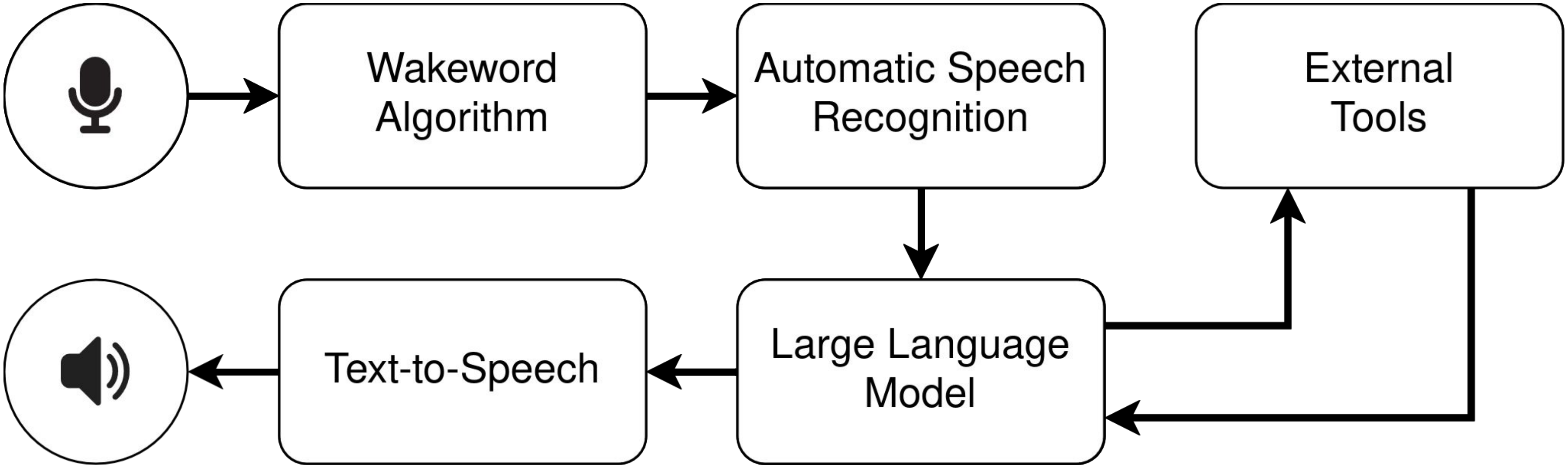
Demo - GPT Backend + Tool Access



```
er token,  3326.60 tokens per second)
[2024-05-06 14:27:48.125] [info] [WASI-NN] llama.cpp: llama_print_timings: prompt eval time =   3929.62 ms /   279 tokens (   14.08 ms p
er token,    71.00 tokens per second)
[2024-05-06 14:27:48.125] [info] [WASI-NN] llama.cpp: llama_print_timings:        eval time =   7408.73 ms /   190 runs   (   38.99 ms p
er token,    25.65 tokens per second)
[2024-05-06 14:27:48.125] [info] [WASI-NN] llama.cpp: llama_print_timings:       total time =  687129.33 ms /   469 tokens
```



System Architecture



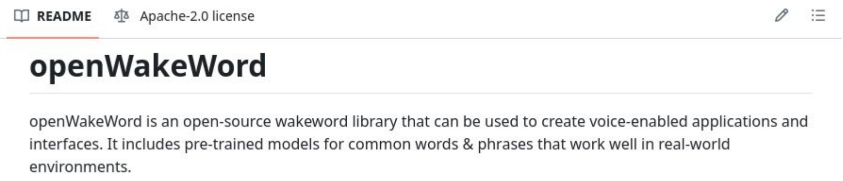
Implementation - Wakeword Algorithm

Wakeword Algorithm

- Detect activation phrase
- Runs all the time
  - Low power
  - Fast response time

openWakeWord

- Open source
- Only 100k model parameters
- Custom phrase with synthetic data
- “Hey Sola!”



openWakeWord

openWakeWord is an open-source wakeword library that can be used to create voice-enabled applications and interfaces. It includes pre-trained models for common words & phrases that work well in real-world environments.

openWakeWord [dscripa]

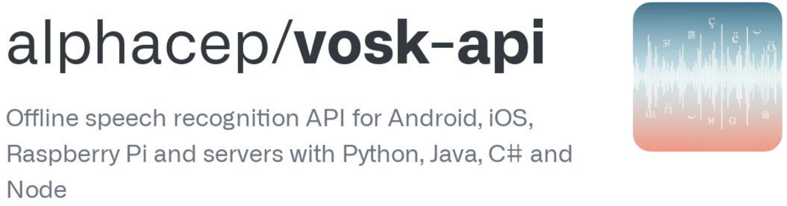
Implementation - Automatic Speech Recognition

Automatic Speech Recognition

- Transcribe text to speech
- Active topic of research
- Support multiple models

OpenAI Whisper

- Well established
- Open source
- Tiny model has only 39 M params



alphacep/**vosk-api**

Offline speech recognition API for Android, iOS, Raspberry Pi and servers with Python, Java, C# and Node

coqui STT

September 21, 2022

Introducing Whisper

We've trained and are open-sourcing a neural net called Whisper that approaches human level robustness and accuracy on English speech recognition.

Implementation - Text-To-Speech Synthesis

Text-to-Speech Synthesis

- Generate audio from text
- Active topic of research
- Support multiple models

Mycroft Mimic 3

- Open source
- Balances quality and compute
- Supports multiple voices

rhasspy/**piper**

A fast, local neural text to speech system

coqui TTS

Implementation - Large Language Model

Less Local ⟷ More Local

ChatGPT API

Cloud Instance

Local Machine

Nvidia Jetson

Implementation - External Tools

How do LLMs Access External Tools?

- Pass natural language command to LLM
- Make LLM output JSON

  {tool: “tool_name”, args: [...arguments]}
- Parse output and execute tool in Python
- Feed result back to LLM

Implemented Tools with Prompt Engineering

- Algebra: Calculate simple equations
- Search: Find Wikipedia Entries
- Weather: Current weather at a location

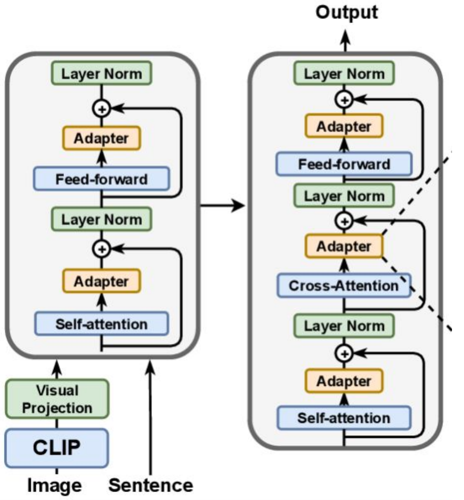
→ **Simple, but powerful proof of concept.**

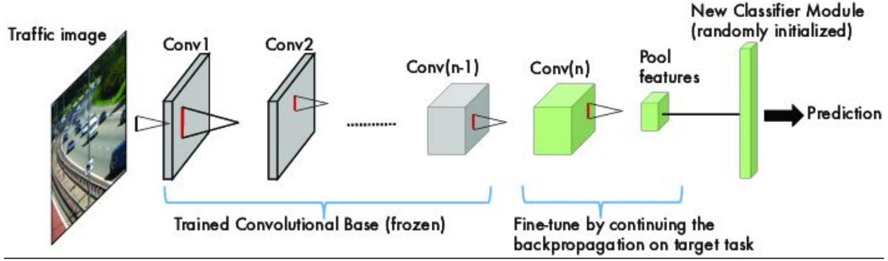
ToolLLM [Qin et al.]

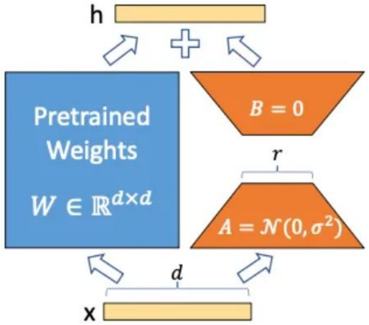
LLM Fine-Tuning

Fine-tuning options:

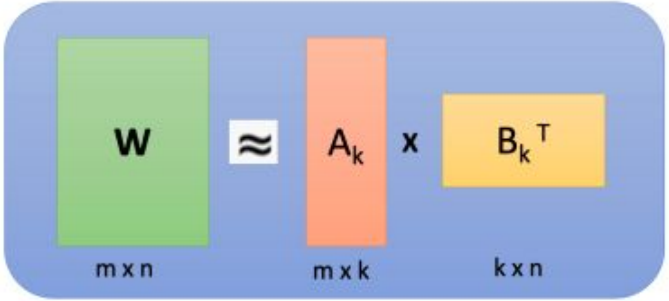
- Projection heads
  - Layer(s) added right before model output
- Adaptors
  - Layer(s) added **between** existing layers of pretrained model
- Low-rank adaptation (LoRA)
  - Layers injected **into** each existing layer of model




Traffic image
Conv1
Conv2
Conv(n-1)
Conv(n)
Pool features
New Classifier Module (randomly initialized)
Prediction
Trained Convolutional Base (frozen)
Fine-tune by continuing the backpropagation on target task


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

LLM Fine-Tuning



LoRA:

- Computes change to original weights
  - Preserves original model, allows swapping between tasks
- Updates low-dim subspace of params
  - Faster adaptation, lower computational cost
- Grants active control over how much fine-tuning affects model behavior through scaling factor

Quantized LoRA:

- Keeps pretrained weights quantized in memory, only LoRA weights are floats
  - Significantly reduces memory required for model training + inference

LLM Fine-Tuning

Quantized LoRA (QLoRA) for fine-tuning:

- Rank = 64 ⟶ controls dimensionality for fine tuning
- alpha = 192 ⟶ contributes to scaling factor for weight update
- Precision = 16-bit ⟶ LoRA weights
- Quantization = 4-bit ⟶ pretrained model weights
- Trained on 4% of data due to time constraints

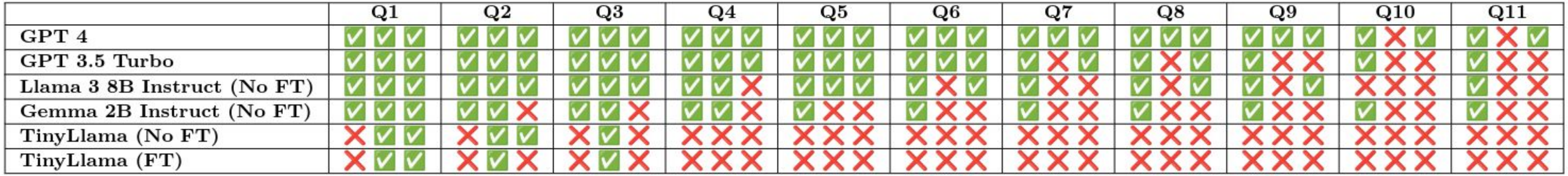
$$W_{\texttt{ft}} = W_{\texttt{pt}} + \frac{\overbrace{\alpha}^{\text{Scaling Factor}}}{r}\underbrace{AB}_{\text{Rank Decomposition Matrix}}$$

Tool-LLM Evaluation

1. What is your name?
2. What can you help me with?
3. Write me a poem about the tools you have access to.
4. What is the current weather in Columbia University, NYC?
5. What is fifty seven multiplied by three hundred and twenty one minus four?
6. In what city did the 2020 olympic games take place?
7. What is the current weather in the city where the 2020 olympic games took place?
8. What is the sum of the current temperature and humidity at Columbia University?
9. What is the product of all the numbers of the year when the Olympic Games were held in Beijing?
10. Write a poem about the current weather in the city where the 2020 olympic games took place.
11. What is the sum of the current temperature and humidity in the city where the 2020 olympic games took place?

Tool-LLM Evaluation Results

Evaluation Results



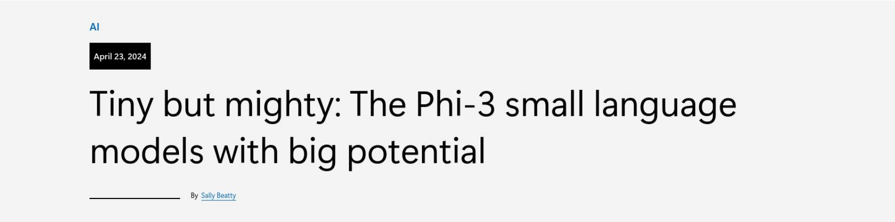
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
GPT 4	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ❌ ✅	✅ ❌ ✅
GPT 3.5 Turbo	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ❌ ✅	✅ ❌ ✅	✅ ❌ ❌	✅ ❌ ❌	✅ ❌ ❌
Llama 3 8B Instruct (No FT)	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ✅	✅ ✅ ❌	✅ ✅ ✅	✅ ❌ ✅	✅ ❌ ❌	✅ ❌ ✅	✅ ❌ ✅	❌ ❌ ❌	✅ ❌ ❌
Gemma 2B Instruct (No FT)	✅ ✅ ✅	✅ ✅ ❌	✅ ✅ ❌	✅ ✅ ❌	✅ ❌ ❌	✅ ❌ ❌	✅ ❌ ❌	✅ ❌ ❌	✅ ❌ ❌	✅ ❌ ❌	✅ ❌ ❌
TinyLlama (No FT)	❌ ✅ ✅	❌ ✅ ✅	❌ ✅ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌
TinyLlama (FT)	❌ ✅ ✅	❌ ✅ ❌	❌ ✅ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌	❌ ❌ ❌

TABLE 2: Tool-LLM evaluation results.
The results indicate (1) if the model produced correct JSON outputs,
(2) whether the correct number of tools was used and (3) if the final answer is correct.
✅ represents a passing result, while ❌ means failing.

- Open source models can be competitive with closed source ones
- Less parameters = Worse performance
- More complex questions = Worse performance

Future Work

- Experiment with new LLMs
- More extensive model fine-tuning
- Better prompt engineering
- Multi-language support
- Multi-modal inputs support
- Access to chat history
- Speaker Recognition
- More tool integrations
- Easy containerized deployment
- …



AI

April 23, 2024

Tiny but mighty: The Phi-3 small language models with big potential

By Sally Beatty

Phi-3 [Microsoft]

Paper | April 2024

OpenELM: An Efficient Language Model Family with Open Training and Inference Framework

Sachin Mehta, Mohammad Sekhavat, Qingqing Cao, Max Horton, Yanzi Jin, Frank Sun, Iman Mirzadeh, Mahyar Najibikohnehshahri, Dmitry Belenko, Peter Zatloukal, Mohammad Rastegari

OpenELM [Apple]

Conclusion

Open source AI is powerful, and can be competitive with commercial solutions.

References

- **Our repository:** https://github.com/eecse6692/e6692-2024Spring-FinalProject-FSCS-fps2116-cs4347
- ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs
- Small LLMs Are Weak Tool Learners: A Multi-LLM Agent
- Toolbench
- openWakeWord
- OpenAI Whisper, Coqui STT, Vosk
- Piper, Coqui TTS, Mimic 3
- OpenAI GPT-3.5, OpenAI GPT-4, TinyLlama, Google Gemma
- Microsoft Phi-3, Apple OpenELM
- LoRA