On-Device Personalized Agent using Fine-tuned Qwen3-0.6B

Brandon Howell
Department Name or Project Group
University or Organization Name
City, Country
Email: your.email@example.com

***Abstract*—This project aims to create a personalized AI assistant capable of running directly on an Android smartphone. The core of this work involves fine-tuning a small large language model (LLM), specifically Qwen3-0.6B, with personal user data to create a private, context-aware agent. This personalized model is then integrated into an experimental Android agent framework, based on Gosling, to enable it to perform tasks by interacting with other applications on the device. We address the significant challenges of on-device LLM inference, user data privacy, and agentic control within the resource-constrained and complex mobile environment. The project explores the trade-offs between full fine-tuning and parameter-efficient fine-tuning (PEFT) techniques to achieve a capable yet efficient personalized agent.**

***Index Terms*—On-Device AI, Small Language Models, LLM Personalization, Agentic AI, Parameter-Efficient Fine-Tuning (PEFT), Android, Qwen3.**

I. Introduction

Scope: This project aims to create a personalized AI assistant capable of running directly on an Android smartphone. The core involves fine-tuning a small, large language model (LLM), specifically Qwen3-0.6B [1], with personal user data and other Android domain specific information. Subsequently, this personalized model will be integrated into an experimental Android agent framework (potentially a fork of Gosling [2]) to enable it to perform tasks by interacting with other applications on the device.

Problem: Current large language models, while powerful, are typically generic and operate in the cloud, which raises privacy concerns and lacks deep personalization. Running large models locally on resource-constrained devices like smartphones is computationally challenging. Enabling these models to interact with the device's applications and data in an automated fashion (i.e., act as agents) presents significant technical hurdles in the mobile environment. This project addresses the need for a private, personalized, and capable AI assistant that lives entirely on a user's device.

Interest & Non-Triviality: This project is interesting because it tackles the intersection of several cutting-edge AI domains:

- *On-Device AI:* Deploying capable LLMs on mobile phones pushes the boundaries of model optimization and efficient inference while preserving privacy.
- *LLM Personalization:* Fine-tuning with personal data explores methods to make AI truly tailored to an individual user's context, habits, and preferences.
- *Agentic AI on Mobile:* Integrating an LLM with device automation frameworks like Gosling explores the practical challenges and potential of creating autonomous agents in the complex and varied Android ecosystem.
- *Parameter-Efficient Fine-Tuning (PEFT):* Experimenting with techniques like adapters offers insights into efficient model adaptation for resource-constrained environments.

The combination of fine-tuning a recent small LLM, deploying it locally, integrating personal context, and building agentic capabilities makes this project ambitious and non-trivial. A conceptual diagram is shown in Fig. 1.

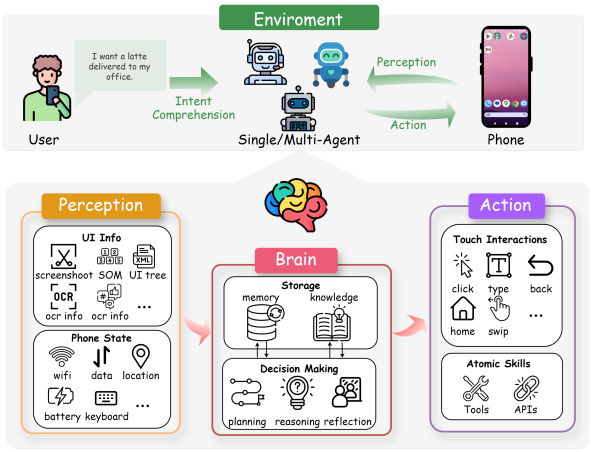

Fig. 1. Conceptual diagram of a mobile LLM agent architecture, where the agent reasons about user tasks and interacts with applications on the device. Image from [3].

II. Related Work

The development of personalized, on-device AI agents draws upon research in several areas.

Small Language Models (SLMs): The trend towards smaller, capable language models is necessary for on-device deployment. Models like Phi-3, Gemma, and the chosen Qwen3-0.6B [1] from Alibaba represent efforts to achieve high performance with fewer parameters. The Qwen3 series specifically notes optimization for resource-constrained environments [4].

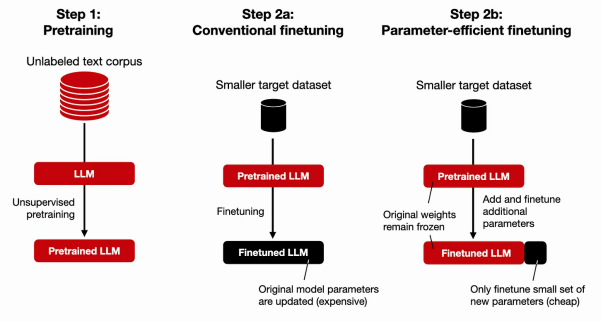

Fig. 2. Illustration of adapter-based PEFT. Small, trainable adapter modules are inserted into a frozen pre-trained model, enabling efficient fine-tuning. Image from [9].

On-Device LLM Inference: Techniques for efficiently running LLMs on mobile devices are critical. This includes model quantization, optimized inference engines (like `llama.cpp`, MediaPipe LLM Inference, ONNX Runtime), and leveraging mobile NPUs/GPUs [5].

LLM Fine-Tuning for Personalization: Adapting pre-trained LLMs is a common practice. While full fine-tuning is resource-intensive, **Parameter-Efficient Fine-Tuning (PEFT)** methods have gained prominence. Techniques like Adapter Tuning [6], LoRA [7], and QLoRA [8] allow adaptation by training only a small number of additional parameters, as illustrated in Fig. 2. This is particularly relevant for on-device scenarios.

LLM-Powered Agents: Frameworks enabling LLMs to use tools and execute tasks are emerging. Applying these concepts to mobile environments involves interacting with apps and system services. Projects like Gosling [2] specifically explore agentic capabilities on Android, using platform features like Accessibility Services.

III. Method

This project is developed in two main phases.

A. Phase 1: Personalized On-Device LLM Fine-tuning

1) *Model Selection:* The core model is Qwen3-0.6B, chosen for its small size and performance [1].
2) *Data Collection & Preparation:* A personal dataset will be curated, including calendar entries, synthesized communication logs, personal notes, app usage patterns, and location habits. The data will be formatted for instruction fine-tuning.
3) *Fine-tuning Implementation:* Two approaches will be explored using Hugging Face `transformers` and `peft`:
   - **Full Fine-tuning:** Update all weights of the model.
   - **Adapter-based PEFT:** Freeze the base model and train lightweight adapter modules [6]–[8].
4) *On-Device Deployment:* The fine-tuned models will be prepared for on-device execution via quantization (e.g., 4-bit GGUF) and deployed using an inference engine like `llama.cpp` with Android bindings.

B. Phase 2: Agentic Integration with Gosling

1) *Framework Setup:* Fork the Gosling Android agent repository and study its architecture for invoking LLMs and interacting with device capabilities.
2) *Model Integration:* Modify the forked Gosling code to use the personalized, locally deployed Qwen3-0.6B model (likely the PEFT version) as its reasoning engine.
3) *Tool Definition & Use:* Explore Gosling's mechanisms for defining and using tools, potentially implementing custom capabilities via a "mobile MCP" provider.
4) *Task Automation:* Define and test simple, multi-step tasks that require the agent to use the personalized LLM and interact with other apps via Gosling's capabilities.

IV. Evaluation

Success will be evaluated based on experimentation depth and insights gained.

A. Phase 1 Evaluation (Personalized LLM)

- *Dataset:* A held-out set of prompts based on the personal data domain.
- *Metrics:*
  - **Qualitative Assessment:** Subjective evaluation of the model's responses for personalization, accuracy, coherence, and tone compared to the base model.
  - **Quantitative Assessment (Exploratory):** Measure perplexity, model storage size, inference latency, and peak memory usage on the Android device.
- *Comparison:* Directly compare the effectiveness and resource trade-offs between the fully fine-tuned and adapter-based models.

B. Phase 2 Evaluation (Agentic Capabilities)

- *Dataset:* A set of defined, multi-step tasks requiring interaction with device features or apps.
- *Metrics:*
  - **Task Completion Rate:** Percentage of tasks successfully completed.
  - **Qualitative Assessment:** Evaluate planning ability, tool usage effectiveness, error handling, and overall usefulness.
  - **Efficiency (Observational):** Note the time and steps required for task completion.

V. Milestones

The project timeline is structured as follows:

- *Week 6:* Download Qwen model, test baseline capabilities, and begin curating the fine-tuning dataset.
- *Week 7:* Fine-tune and test the personalized model. Evaluate its qualitative and quantitative improvements over the base model.
- *Week 8:* Fork the Gosling repository and begin integration work for the agent framework.
- *Week 9:* Combine the fine-tuned Qwen3 with Gosling to automate tasks. Test and gather quantitative results on task success.

- *Week 10:* Finalize features, complete testing and evaluations, and prepare the final report.

VI. Task Assignments

This is a solo project, so all tasks will be handled by the author.

References

[1] Qwen Team, "Qwen3-0.6B," Hugging Face, 2024, [Online]. Available: https://huggingface.co/Qwen/Qwen3-0.6B.

[2] Block, Inc., "Gosling: A framework for building agents on mobile," GitHub Repository, 2024, [Online]. Available: https://github.com/block/gosling.

[3] W. Shi et al., "Unleashing the Power of Mobile LLMs: A Survey on Frontiers, Applications, and Challenges," Preprints.org, 2024, [Online]. Available: https://www.preprints.org/manuscript/202401.1278/v1.

[4] Qwen Team, "Qwen3 LLM Series: A Big Leap in Performance, Tokenization, and Multilingual Capabilities," QwenLM Blog, June 2024, [Online]. Available: https://qwenlm.github.io/blog/qwen3/.

[5] A. Kim et al., "MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases," arXiv preprint arXiv:2312.03863, 2023.

[6] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-Efficient Transfer Learning for NLP," in Proc. Int. Conf. on Machine Learning (ICML), 2019.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in Proc. Int. Conf. on Learning Representations (ICLR), 2022.

[8] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," arXiv preprint arXiv:2305.14314, 2023.

[9] S. Raschka, "Finetuning LLMs with Adapters," Sebastian Raschka's Magazine, 2023, [Online]. Available: https://magazine.sebastianraschka.com/p/finetuning-llms-with-adapters.