MedDialogue: General-Purpose Medical Dialogue Fine-Tuning Framework

Project Information

Lead/Mentor: Frederick Gyasi

Contributors: Frederick Gyasi

Institution: Medical University of South Carolina, Biomedical Informatics Center, Clinical NLP Lab

Current Funding: N/A

Future Funding: N/A

IRB #: N/A

RMID: N/A

SPARCRequest: N/A

Project Summary

MedDialogue is a general-purpose framework for fine-tuning large language models (LLMs) on ANY medical dialogue task through conversational learning. The framework is task-agnostic and contains no domain-specific code - users define their own tasks, output fields, and question templates.

Unlike traditional instruction fine-tuning that produces brittle models failing on question variations, MedDialogue trains models through natural multi-turn clinical dialogues, enabling robust handling of varied questions, follow-ups, and context.

Key Research Questions

1. **Can conversational fine-tuning produce more robust medical AI models than traditional instruction fine-tuning?**

2. **How effective are multi-turn dialogue patterns in teaching clinical reasoning compared to single-turn approaches?**

Framework Capabilities

Core Architecture

Component	Description
Task Definition	User-defined via `TaskConfig` (task name, input field, output fields, question templates)
Conversation Generation	Configurable via `ConversationConfig` (turn ratios, typos, validation split)
Question Combination	16 styles: 7 grammatical variations + 9 logical reasoning patterns
Output Formats	TEXT, JSON, XML, Markdown with configurable weighted ratios
Training Method	LoRA/QLoRA fine-tuning with Unsloth optimization
Data Mapping	1:1 mapping (each clinical note generates ONE training conversation)

Supported Models

Model Family	Variants	Max Sequence Length
LLaMA	3.1, 3.2 (8B, 70B)	32,768 tokens
Phi	Phi-4 (14B)	16,384 tokens
Mistral	7B, Nemo	32,768 tokens
Qwen	2.5 (7B, 14B, 32B)	32,768 tokens

Question Generation Styles

Category	Styles	Description
Grammatical (1-7)	Sequential, Conjunctions, Transitional, Numbered, Natural, Directive, Mixed	Work with any question order
Logical (8-16)	Causal, Conditional, Prioritized, Comparative, Temporal, Hierarchical, Dialectical, Exploratory, Dependency	Require ordered questions for reasoning flow

---

Example Use Case: Pediatric Malnutrition Assessment

The repository includes a demonstration using pediatric malnutrition assessment (ASPEN/WHO/CDC guidelines) to show how to use MedDialogue:

File	Purpose
`train_malnutrition.py`	Training script demonstrating TaskConfig setup
`evaluate_malnutrition.py`	Evaluation script with classification metrics
`malnutrition_system_prompts.py`	Domain-specific system prompts

This example demonstrates: - 11 clinical output fields - 110 question templates (10 per field) - Field ordering following clinical logic (assess, diagnose, plan) - Multi-turn clinical dialogue patterns

---

Resources

Code Repository

GitHub: https://github.com/gyasifred/meddialogue

Core Framework Modules

Module	Function
	Main `MedDialogue` class - high-level training interface

Module	Function
`meddialogue/core.py`	
`meddialogue/config.py`	Configuration dataclasses (`TaskConfig` , `ConversationConfig` , `TrainingConfig` , `LoRAConfig` , `ModelConfig`)
`meddialogue/data_prep.py`	`DataPrep` class - conversation generation with 16 question styles
`meddialogue/train.py`	`Trainer` class - training pipeline with Unsloth
`meddialogue/inference.py`	`InferencePipeline` - single/multi-turn inference
`meddialogue/models.py`	`ModelRegistry` - model loading and LoRA application
`meddialogue/utils.py`	Text processing, output formatting, parsing utilities
`meddialogue/safety.py`	Safety stubs for backward compatibility

Data Requirements

Users provide CSV data with:

Required	Description
Input field	Clinical text column (configurable, default: `clinical_note`)
Output fields	User-defined columns matching `TaskConfig.output_fields`

Acknowledgements

Lead/Mentor: Frederick Gyasi (gyasi@musc.edu)

Contributors: Frederick Gyasi

Past Contributors: N/A

Institution: Medical University of South Carolina, Biomedical Informatics Center, Clinical NLP Lab

Current Funding: N/A

Future Funding: N/A

License: MIT

Technical Dependencies

Package	Purpose
Unsloth	Fast LLM fine-tuning with memory optimization
Transformers	Hugging Face model infrastructure
PEFT	Parameter-efficient fine-tuning (LoRA)
PyTorch	Deep learning framework
Datasets	HuggingFace dataset handling
pandas	Data processing
scikit-learn	Evaluation metrics

Contact: gyasi@musc.edu

Medical University of South Carolina, 2025