Multi-Action Voice Chatbot System Report

Project Objective

As part of online assessment for “AI Engineer internship at Agile Loop”, The task was to develop a voice-enabled chatbot system capable of handling multiple analysis tasks in parallel through intent detection. The system processes voice input, detects intent, executes multiple relevant analyses simultaneously, and synthesizes a comprehensive response. The implemented solution is complete, robust and fast as well.

Key Features

1. **User Query Intent Detection**
    - Custom-trained spaCy textcat multilabel classifier
    - Handles 5 classes (classify, factcheck, summarize, analyze, detail)
    - Lightweight and quick
    - Dynamic intent scoring and routing
2. **Real-time Voice input**
    - Whisper large-v3 model integration
    - 16kHz sampling rate with configurable recording duration
    - Real-time audio quality checks and silence detection
    - Could have tried canary-1b (nvidia) but couldn't due to time constraints
3. **Parallel Task Execution**
    - LangGraph-based parallel workflow execution
    - Dynamic task routing based on intent confidence
    - Content-aware task selection
4. **Context Management**
    - LangGraph memory-based state management
    - Thread-safe conversation history tracking
    - Stateful parallel execution handling
    - Efficient message accumulation using reducers

Technical Challenges & Solutions

1. **BERT models fewshot**
    - Used base-BERT, RoBERTa and even ModernBERT for text classification (using fewshot examples) but the predictions were worse than random guess
    - Poor performance on specialized intents like 'factcheck' and 'summarize'

Tech Stack/Core Components

- **Query intent detection**: spaCy (textcat multilabel)
- **STT**: Whisper large-v3
- **LLM**: Qwen 2.5 3B
- **Framework**: LangGraph/LangChain

Code files explanation

1. (`base.py`): abstract chatbot foundation with essential methods for response generation and logging.
2. (`query_classifier.py`): Custom-trained spaCy multilabel classifier for intent detection with a lightweight model trained on 10+ examples per intent.
3. (`stt_hf.py`): Whisper large-v3 implementation handling real-time audio capture and transcription with 16kHz sampling and attention masking.
4. `(main.py`): integrates user query intent detection and speech processing through a dynamic workflow managing five specialized nodes (classify, factcheck, summarize, analyze, detail) with parallel execution capabilities. The system uses Qwen2.5-3b for response generation with specialized prompting templates, and implements a fan-out/fan-in architecture for task execution - all orchestrated through a thread-safe memory system (and reducer operations for parallel task handling)
5. (`training-spacy-classifier.ipynb`): Custom training setup for spaCy textcat with domain-specific examples.