CDW Documentation

Lie Detector AI Web Application – Documentation

Lie Detector AI Web Application – Documentation

Overview

The Lie Detector AI is a Flask-based web application that:

- Accepts an audio file from the user
- Transcribes the speech using Azure Speech Services
- Analyzes sentiment using Azure Language Service
- Detects deception using Azure OpenAI (GPT-4.1)
- Displays results including transcript, sentiment, and AI-generated lie detection analysis

It integrates with Azure Monitor, Application Insights, and Log Analytics for telemetry, diagnostics, tracing, and alerting.

Summary: What the App Does

The application allows users to upload an audio file containing speech. It processes the audio by transcribing it to text using Azure Cognitive Services. The resulting transcript is analyzed for sentiment (positive, neutral, negative), and then evaluated by a GPT-4.1 model deployed in Azure OpenAI to determine whether the statement is likely a truth or a lie.

All stages of processing are monitored with Application Insights and emit telemetry signals for logging, errors, and latency.

Order of Events (Request Processing Flow)

1. **User Submits Audio File**
   - User uploads a .wav file via the web interface (`/` route, POST method).
   - File is saved locally as `temp.wav`.

1. **Speech-to-Text Transcription**

- Azure Speech SDK is configured using credentials from Azure Key Vault.
- The file is transcribed using continuous speech recognition.
- Recognized text segments are collected and merged into a single transcript.
- If the transcript is empty, an error is logged, and the user is notified.

1. **Sentiment Analysis**

- The transcript is passed to Azure Language Service (Text Analytics API).
- The API returns the sentiment (positive, neutral, or negative) and associated confidence scores.
- A telemetry event is logged with the sentiment analysis results.

1. **Lie Detection via GPT-4.1**

- The transcript is sent to a deployed GPT-4.1 model via the Azure OpenAI API.
- A structured prompt instructs the model to determine whether the statement is truthful or deceptive.
- The response is parsed, logged, and returned to the front end.

1. **Rendering the Result**

- The final HTML page displays:
    1. The transcribed statement
    2. Sentiment and confidence levels
    3. AI's lie detection result

Telemetry and Logging

Throughout the process, custom telemetry is sent to Application Insights using:

- `log_event()` for success states
- `log_error()` for handled exceptions
- `trace_operation()` for performance monitoring of major steps like sentiment analysis and lie detection

Architecture

Component	Service
App Backend	Flask
Speech-to-Text	Azure Speech Service
Sentiment Analysis	Azure Language Service
Lie Detection (LLM)	Azure OpenAI (GPT-4.1)
Secrets Management	Azure Key Vault
Monitoring & Logging	Azure Application Insights, Log Analytics
Alerts	Azure Monitor
Dashboard	Azure Portal (Custom Metrics/Logs Dashboard)

Features

- Upload audio and extract spoken text
- Detect sentiment: positive, neutral, or negative
- Analyze deception with GPT-4.1
- Log all actions to Azure Monitor
- Real-time error and latency alerts (threshold: 500ms)
- Visual telemetry dashboard for client insights

Monitoring and Telemetry

Application Insights

- Logs performance metrics (requests, dependencies, failures)
- Tracks custom events such as transcription results and lie detection outputs
- Uses OpenCensus for trace-level logging and distributed tracing

Alerts

- Triggered if OpenAI response latency exceeds 500 milliseconds
- Triggered on any uncaught exceptions or application errors
- Alerts are managed and visualized via Azure Monitor

Dashboard

- Custom dashboard built in Azure Portal
- Displays request volume, sentiment trends, AI response latency, and failure rates
- Includes charts, metrics, and log query visualizations

Security

- Secrets are stored in and retrieved from Azure Key Vault
- No sensitive credentials are hardcoded
- HTTPS and secure Azure resource access are enforced

Notes

- While the funcionality of the app works according to this documentation, it does not work well as a lie detection app. I learned that sentiment analysis doesn't really help the lie detection GPT at all.
- Most of the responses it was giving me were neutral, unless I said something that was factually incorrect.
- It works more like a fact-checker than a lie detector, after testing.