MindTrace AI-Powered Mentor Evaluation System

Technical Summary: MindTrace AI-Powered Mentor Evaluation System

1. Problem Statement

Traditional methods of evaluating teaching and mentorship are often subjective, labor-intensive, and lack actionable granularity. Human evaluators may struggle to provide consistent, evidence-based feedback across multiple dimensions (e.g., clarity, pacing, checking for understanding) or pinpoint the exact moments where an explanation fails. Furthermore, providing constructive alternatives (rewrites) or detecting subtle logical inconsistencies across a long session is cognitively demanding and time-consuming. MindTrace aims to automate this process, providing an explainable, scalable, and multi-dimensional analysis of teaching sessions using advanced Large Language Models (LLMs).

2. Approach and AI Components

MindTrace employs a multi-stage AI pipeline designed to mimic and augment expert human evaluation. The system moves beyond simple sentiment analysis to perform deep pedagogical assessment.

- **Hybrid LLM Strategy:** The system utilizes a hybrid approach, leveraging Google Gemini 2.5 Flash as the primary model for its multimodal capabilities and speed, with Groq LLaMA 3.3 70B as a secondary option for high-performance text inference.
- **Automated Transcription:** Video content is processed using Google Gemini 1.5 Flash, which generates timestamped transcripts directly from video files, ensuring alignment between spoken words and temporal segments.
- **Multi-Dimensional Evaluation:** The core evaluation engine analyzes segments against 10 distinct metrics:
  - **Core:** Clarity, Structure, Correctness, Pacing, Communication.
  - **Advanced:** Engagement, Examples, Questioning, Adaptability, Topic Relevance.
  - The LLMEvaluator service uses complex prompts to force the LLM to output structured JSON data containing scores (1-10) and detailed reasoning for each metric.
- **Evidence Extraction:** To solve the "black box" problem of AI, the EvidenceExtractor locates specific problematic phrases (5-30 words) within the transcript. It calculates exact character start/end positions to highlight issues like "vague terminology" or "logical gaps" directly in the user interface.
- **Generative Improvement (Rewriting):** The ExplanationRewriter detects low-clarity segments (score < 7.0) and generates improved versions. It aims to increase the word count by 20-50% to enhance clarity, using prompts that target specific deficiencies like structure or engagement.
- **Coherence Analysis:** The CoherenceChecker analyzes the full session context to detect session-wide issues, specifically looking for contradictions, topic drift, and logical gaps between segments that isolated segment analysis might miss.

3. Technical Architecture

The application follows a modern decoupled architecture, separating the heavy AI processing from the user interface.

Backend (FastAPI + Python)

- **Framework:** FastAPI is used for high-performance async request handling, essential for long-running AI tasks.
- **Service Layer Pattern:** The codebase is organized into distinct services (transcription.py, llm_evaluator.py, evidence_extractor.py, coherence_checker.py), ensuring modularity and easier testing.
- **Data Persistence:** MongoDB (accessed via Motor async driver) stores user profiles, session metadata, and complex nested evaluation results.
- **Configuration:** A robust config.py manages environment variables, allowing dynamic switching of LLM strategies ("gemini", "groq", or "hybrid") and scoring weights (e.g., Clarity weighted at 0.25).

Frontend (React)

- **Framework:** React 18 with TailwindCSS for styling.
- **Visualization:** Uses Recharts and D3.js for rendering performance charts and "Explanation Flow Graphs" to visualize teaching quality over time.
- **Deployment:** The frontend is deployed on Vercel, while the backend resides on Hugging Face Spaces to leverage GPU/inference capabilities if needed.

4. Challenges and Mitigations

Challenge	Technical Mitigation
Hallucination of Evidence	The EvidenceExtractor validates that every extracted "problematic phrase" actually exists in the source text before returning it. It uses strict string matching to ensure the UI highlights real text rather than hallucinated quotes.
LLM Output Instability	The system enforces strict JSON response formats in prompts. The LLM Evaluator includes validation logic that checks for required keys (e.g., 'clarity', 'score') and falls back to a mock evaluation if the LLM output is malformed, ensuring system stability.
Subjectivity of "Relevance"	The prompting strategy explicitly instructs the LLM to value "related topics" that enhance understanding (scoring them 8-10) and only penalize completely unrelated tangents, preventing valid teaching analogies from being flagged as "off-topic".
Processing Latency	The architecture supports asynchronous processing. The transcription service polls the Gemini API for file processing status (PROCESSING -> ACTIVE), preventing timeouts on large video files.

5. Roadmap to Final Build

01

Phase 1: Core Foundation (Completed)

- Implementation of user auth (Firebase), video upload, and basic mentor management.
- Deployment of the transcription pipeline and core 5-metric evaluation system.

02

Phase 2: Deep Analytics (In Progress/Beta)

- **Evidence Extraction:** Logic is implemented to pinpoint exact character indices of errors.
- **Smart Rewrites:** The backend now supports generating specific "Before/After" comparisons with confidence scoring.
- **Coherence Checking:** Algorithms for detecting contradictions and logical gaps are implemented in the backend.

03

Phase 3: Advanced & Enterprise (Planned)

- **Predictive Insights:** ML-based predictions for mentor improvement trajectories.
- **Reporting:** Generation of PDF reports for stakeholders.
- **Real-time Analysis:** Streaming video analysis for live feedback.
- **Accessibility:** WCAG 2.1 Level AA compliance and mobile optimization.