Intelliview

A Personalized AI Interview Coach with Fine-Tuned RAG and Multimodal Analysis

Team Members:
Chih-Yuan Chang, Ching-Heng Huang, Chung-Yeh Yang, Nan Tang Wu

Abstract

Technical job interviews pose a unique challenge: candidates must dynamically connect their past experiences to Job Description (JD) requirements through contextual questions. Standard Large Language Models (LLMs) struggle with this task, often producing generic responses or hallucinating details when asked to retrieve specific resume bullet points that best answer interview questions. To address this retrieval challenge, we present Intelliview, an end-to-end AI-powered interview coaching system. At its core, Intelliview employs a fine-tuned Sentence Transformer model trained via Supervised Contrastive Learning (MultipleNegativesRankingLoss) to align the embedding space where queries composed of "JD + Interview Question" map directly to the most relevant "Resume Bullet Points." We fine-tuned two base models—all-MiniLM-L6-v2 and all-mpnet-base-v2—on a dataset of 5,000 (JD, Question, Bullet) triplets. Our best-performing model (Fine-Tuned MPNet) achieved **Hit@1 of 49.8%** and **Hit@5 of 75.3%**, significantly outperforming baseline embeddings. The system is deployed as a FastAPI web application featuring a complete pipeline: (1) PDF resume parsing with structured information extraction, (2) semantic retrieval using the fine-tuned embedding model, and (3) a dual-mode mock interview engine integrating OpenAI and Gemini APIs for natural language generation. Additionally, we incorporate Computer Vision analysis using MediaPipe for multimodal performance assessment, tracking non-verbal cues including gaze patterns, facial expressions, posture stability, and gesture naturalness—providing candidates with holistic feedback on both verbal content and professional presence.

Project Resources:
Code Repository: `https://github.com/EdChang716/Intelliview-AML_project`
Demo Video: `https://youtu.be/g0IjEFeOzBE?si=AqNGlzOBdSLS7VK-`

1 Introduction & Motivation

Preparing for technical interviews is a high-stakes challenge. Candidates often struggle not with a lack of technical skills, but with the ability to *contextualize* their past achievements to fit specific job requirements. Existing tools generally fall into two categories: static question lists that lack personalization, or generic LLM chatbots (e.g., standard ChatGPT) that lack deep knowledge of the user's specific history.

A critical limitation in vanilla RAG (Retrieval-Augmented Generation) systems is the "semantic gap." Standard embedding models are trained on general similarity, not on the logic of hiring. For example, when asked "Tell me about a time you optimized a system," a standard model might retrieve a generic sentence about coding, rather than the specific bullet point mentioning "Reduced latency by 40% using Redis."

To address this, we propose **Intelliview-AML**. Our core hypothesis is that by fine-tuning embedding models to treat (JD + Question) as the query and the Resume Bullet as the document, we can significantly reduce hallucinations and ground the AI's feedback in the user's actual, most relevant experience.

2 Related Work & Inspiration

Our approach was inspired by a critical gap in the current market of career coaching tools. While widely used platforms like **VMock** excel at syntactic checks—such as formatting alignment and keyword matching—we observed that they fail to provide semantic coaching. They lack the ability to judge if a user's specific experience logically answers a dynamic situational question.

For inspiration on bridging this gap, we looked to emerging AI coaching platforms such as `https://www.mbbcase.help/`. This tool demonstrates how AI can facilitate interactive case studies with adaptive feedback. Motivated by this capability, we aimed to engineer a similar depth of reasoning for general technical interviews, moving beyond static Q&A to a system that understands the "logic" of a candidate's past experience.

To achieve this, our work intersects with three key technical areas:

- **Retrieval-Augmented Generation (RAG) for Recruitment:** We extend standard RAG pipelines by optimizing the retrieval step specifically for recruitment data. Unlike general QA systems, our retrieval target is highly personal and context-dependent.
- **Contrastive Learning:** Leveraging architectures like Sentence-BERT, we utilize *MultipleNegativesRankingLoss*. This technique is crucial for domain adaptation, forcing the model to push the embeddings of specific questions closer to their corresponding correct resume answers while distancing them from irrelevant experiences.
- **Multimodal Affective Computing:** While previous works focused primarily on text or simple audio signals, our system advances the field by combining deep semantic content analysis with visual behavioral metrics (gaze, movement) for a comprehensive coaching experience.

3 Methodology

The Intelliview system consists of three main stages: Data Synthesis, Model Fine-Tuning, and the FastAPI Application Pipeline.

3.1 Dataset Construction (Synthetic Generation)

Since no public dataset maps “Resume Bullets” to “Contextual Interview Questions,” we constructed a high-quality synthetic dataset to supervise our training:

1. **Source Data:** To ensure both general applicability and domain-specific depth, we aggregated data from two primary Kaggle sources:
   - *Jobs and Job Description Dataset*[1]: Used to provide a diverse range of general role descriptions.
   - *Data Scientist Jobs Dataset*[2]: Specifically included to enhance the model’s understanding of niche technical terminology in Data Science and Machine Learning roles.
2. **LLM-as-Interviewer:** Using GPT-4o-mini, we simulated an interviewer persona. For each resume bullet, the LLM was tasked to generate a specific, probing interview question based on the provided JD.
3. **Triplet Generation:** This process produced training pairs consisting of:
   - *Anchor (Query):* The Job Description text + The generated Interview Question.
   - *Positive (Document):* The specific Resume Bullet Point that answers the question.
   - *Negatives:* In-batch negatives (other random bullets from the batch).

3.2 Model Fine-Tuning Strategy

We trained and evaluated two distinct models to explore the trade-off between inference latency and retrieval accuracy:

- **Model A: all-MiniLM-L6-v2:** A lightweight model optimized for speed and low-resource environments.
- **Model B: all-mpnet-base-v2:** A larger, more powerful model known for capturing subtle semantic nuances.

Both models were fine-tuned using the **SentenceTransformers** framework with *MultipleNegativesRankingLoss* for 10 epochs. This loss function was chosen because it effectively maximizes the similarity between the (JD + Question) and the correct Bullet, teaching the model the "logic" of relevance in an interview context.

[1] https://www.kaggle.com/datasets/kshitizregmi/jobs-and-job-description

[2] https://www.kaggle.com/datasets/andrewmvd/data-scientist-jobs

4 System Implementation & Architecture

The core of our project is a robust web application built with a **FastAPI** backend, designed to orchestrate the flow between user data, the retrieval engine, and external LLM APIs (Figure 1).

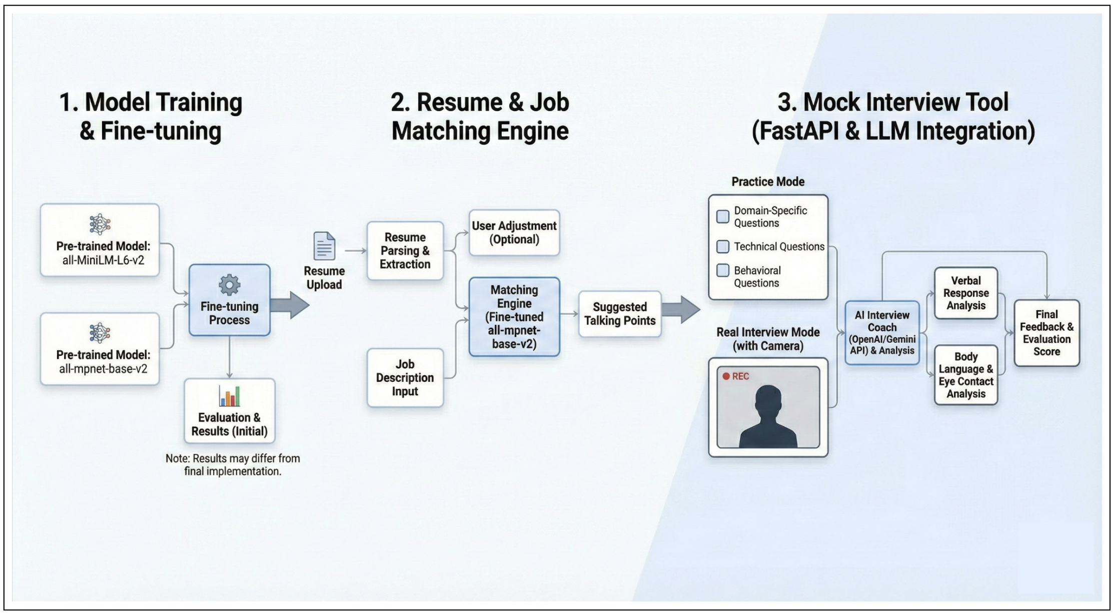

1. Model Training & Fine-tuning
2. Resume & Job Matching Engine
3. Mock Interview Tool (FastAPI & LLM Integration)
Pre-trained Model: all-MiniLM-L6-v2
Pre-trained Model: all-mpnet-base-v2
Fine-tuning Process
Evaluation & Results (Initial)
Note: Results may differ from final implementation.
Resume Upload
Resume Parsing & Extraction
User Adjustment (Optional)
Matching Engine (Fine-tuned all-mpnet-base-v2)
Job Description Input
Suggested Talking Points
Practice Mode
Domain-Specific Questions
Technical Questions
Behavioral Questions
Real Interview Mode (with Camera)
REC
AI Interview Coach (OpenAI/Gemini API) & Analysis
Verbal Response Analysis
Body Language & Eye Contact Analysis
Final Feedback & Evaluation Score

Figure 1: The Intelliview System Pipeline. The system integrates fine-tuned retrieval with multimodal analysis APIs.

4.1 Resume Onboarding & Parsing

The first step in the user journey is digitizing their experience.

- **Parsing:** Users upload their resume in PDF format. The system employs a parsing module to extract raw text and intelligently segment it into individual bullet points.
- **Human-in-the-Loop:** We implemented an editing interface where users can review the extracted bullets. This allows them to fine-tune or correct extraction errors before the data is vectorized, ensuring the database contains high-quality "ground truth."

4.2 Contextual Retrieval Engine

Once the resume is stored, the user inputs a target Job Description. This triggers our core retrieval mechanism:

1. The system combines the target JD with the current interview question.
2. This combined query is encoded by our **Fine-Tuned MPNet**.
3. We perform a cosine similarity search against the user's resume vector store to retrieve the top-k most relevant bullet points. This ensures the AI coach has the exact context needed to guide the user.

4.3 Mock Interview: Dual-Mode Engine

We designed two distinct modes to cater to different preparation stages. In both modes, we integrate **OpenAI** and **Gemini APIs** to act as the "Interviewer," generating questions and providing feedback.

4.3.1 Practice Mode (Targeted Training)

In this mode, the user has full control. They can select specific categories (e.g., *Technical Questions*, *Behavioral Questions*, *System Design*). The AI generates a question, the user types or speaks an answer, and the system provides immediate text-based feedback on content relevance, suggesting which resume bullet they should have highlighted.

4.3.2 Real Mode (Multimodal Simulation)

This mode simulates a realistic, high-pressure mock interview with a dynamic AI interviewer.

- **Structured Interview Flow:** Powered by OpenAI's Large Language Models (LLM), the system generates a complete mock interview session, including introduction, behavioral, project deep-dive, and follow-up questions. The progression is adaptive, depending on the candidate's responses, remaining time, and interview configuration.
- **Interviewer Persona & Voice:** Each session includes a configurable interviewer persona (role, style, and tone). Leveraging OpenAI's LLM for dynamic text generation, questions are tailored to the persona and delivered via text-to-speech to create a realistic interview experience that resembles interaction with a human interviewer.
- **Verbal Analysis:** The backend transcribes the user's spoken responses and performs multi-level verbal analysis using large language models and speech signals. This includes evaluation of:
  - **Content quality:** clarity, logical structure (e.g., STAR), relevance to the job description, and depth of reasoning.
  - **Speech delivery:** speaking rate, pauses, pitch variation, and fluency, which serve as indicators of confidence, hesitation, and communication effectiveness.

  Based on this analysis, the system generates interviewer-style reactions and adaptive follow-up questions in real time.
- **Non-Verbal Analysis (Computer Vision – Extension):** The framework is designed to support non-verbal cue analysis through computer vision, such as:
  - **Gaze detection:** monitoring eye contact with the camera versus looking away.
  - **Body movement analysis:** detecting excessive fidgeting or nervous movements.

  These signals can be incorporated to provide additional feedback on presentation fluency and interview presence.

5 Evaluation & Results

5.1 Quantitative Retrieval Performance

We evaluated the models on a hold-out test set using **Hit@k**. The results (Table 1) clearly demonstrate the impact of fine-tuning.

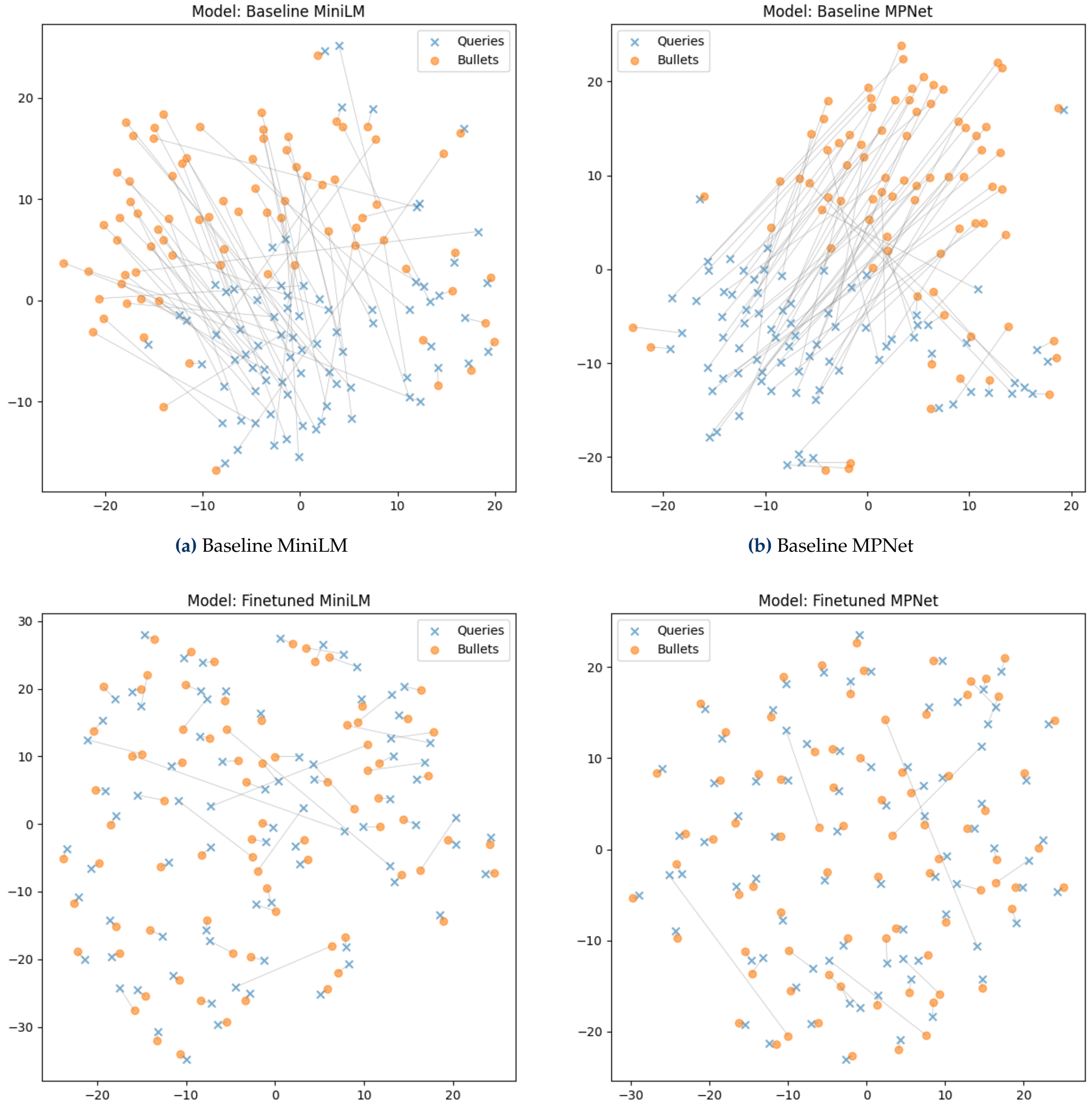
The **Fine-Tuned MPNet** achieved a Hit@1 of 49.8%, effectively doubling the performance of the baseline. This validates that standard models struggle to understand the specific "JD-to-Resume" relationship without domain-specific training.

Table 1: Performance Comparison: Baseline vs. Fine-Tuned Models

Metric	Base MiniLM	Base MPNet	FT MiniLM (Ours)	FT MPNet (Ours)
Hit@1	21.5%	21.6%	39.0%	**49.8%**
Hit@3	32.0%	32.3%	55.3%	70.0%
Hit@5	36.9%	40.2%	61.8%	**75.3%**

5.2 Embedding Space Analysis

Figure 2 visualizes the t-SNE projection of the embedding spaces. The fine-tuned models show significantly tighter clustering of relevant question-answer pairs compared to the scattered distribution of the baselines.



(a) Baseline MiniLM

(b) Baseline MPNet

(c) Fine-Tuned MiniLM

(d) Fine-Tuned MPNet

Figure 2: t-SNE visualization comparing embedding spaces. The fine-tuned models (bottom) show distinct clustering structure, indicating better alignment between JDs and Resume Bullets.

6 Conclusion & Future Work

6.1 Conclusion

This project presented Intelliview, a novel end-to-end system designed to bridge the semantic gap between abstract job descriptions and concrete personal experiences. Unlike generic LLM chatbots that often provide generalized advice or suffer from hallucinations, our system leverages a domain-specific retrieval engine fine-tuned via supervised contrastive learning.

The experimental results, specifically the significant leap in Hit@1 accuracy from 21.6% to 49.8%, emphatically confirm our core hypothesis: off-the-shelf embedding models lack the specific reasoning required for recruitment tasks, and targeted fine-tuning is essential for high-stakes retrieval applications. Furthermore, by seamlessly integrating computer vision for non-verbal analysis alongside the textual RAG pipeline, Intelliview moves beyond simple question-answering. It offers a holistic coaching experience that simultaneously addresses the *content relevance* of the candidate's answer and the *professionalism* of their delivery.

6.2 Future Work

Looking ahead, several avenues exist to further enhance the system's capabilities and robustness:

- **Reinforcement Learning from Human Feedback (RLHF):** We plan to implement a feedback loop where users can explicitly rate the relevance of the retrieved resume bullets during practice sessions. This user-generated signal will serve as a continuous data flywheel, allowing us to iteratively refine the retriever beyond the initial synthetic dataset.
- **Model Optimization and Distillation:** While the Fine-Tuned MPNet offers superior accuracy, its inference latency can be a bottleneck in real-time video modes. Future work will explore knowledge distillation techniques to compress the performance of MPNet into a lighter architecture (like MiniLM) without sacrificing retrieval precision.
- **Multimodal Fusion:** Currently, the text and video analyses operate as parallel streams. We aim to develop a fused multimodal model that can correlate specific physiological cues (e.g., a drop in gaze confidence) with specific parts of the spoken answer (e.g., when discussing a technical weakness), providing even deeper behavioral insights.

7 Contributions

Team Member	Key Responsibilities & Contributions
Chih-Yuan Chang	Data synthesis pipeline, model fine-tuning experiments, performance evaluation, workflow design and web app prototype development
Ching-Heng Huang	Re-built model fine-tuning experiments (MPNet/MiniLM), integrated Gemini API for dynamic questioning, and optimized the resume parsing pipeline. Constructed project report and slide.
Chung-Yeh Yang	Designed and enhanced frontend(UI/UX), developed and optimized resume parser; Re-built the realistic mode pipeline
Nan Tang Wu	LLM prompt engineering (OpenAI/Gemini), real-time interview logic, report writing, and QA testing.

References

[1] Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. EMNLP 2019.

[2] Lewis, P., et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. NeurIPS 2020.

[3] Song, K., et al. (2020). *MPNet: Masked and Permuted Pre-training for Language Understanding*. NeurIPS 2020.

[4] Gao, T., et al. (2021). *SimCSE: Simple Contrastive Learning of Sentence Embeddings*. EMNLP 2021.