**Real-Time Interview Screening System: Post-Processing and System Integration**

**Abstract**

This report details my contributions to a real-time interview screening system developed as part of my Master's dissertation. The project focused on creating a system capable of conducting realistic interview simulations with candidates, leveraging AI for both real-time interaction and comprehensive post-interview analysis. My primary focus areas were the post-processing pipeline, specifically speech-to-text conversion, user interface design, answer accuracy verification, and audio sentiment analysis.

**1. Introduction**

This project aimed to build an AI-driven interview screening system to automate and enhance the candidate evaluation process. The system engages candidates in a dynamic interview experience, utilizing real-time speech processing to guide the conversation flow. My role centered on the post-interview analysis, extracting valuable insights from both the textual content and the emotional nuances of the interview.

A diagram of a diagram

Description automatically generated

Figure 1. System Workflow

The diagram illustrates the system's overall architecture. My contributions primarily reside within the "Post Processing" section, along with key aspects of the real-time processing.

**2. Speech-to-Text Conversion**

**Challenge**: Achieving accurate and cost-effective transcription of candidate speech was essential for subsequent analysis. OpenAI's Whisper API, while robust, presented a budgetary constraint due to its paid nature.

**Solution**: I leveraged the open-source availability of Whisper, opting for a locally hosted implementation ([1]). This approach, while demanding greater computational resources (necessitating GPU acceleration), provided free and reliable multi-lingual transcription capabilities. The "medium" model proved sufficiently accurate for our purposes.

**3. User Interface Design**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Challenge**: Creating a user-friendly and responsive interface was crucial for facilitating smooth candidate interaction, parameter configuration, and system control.

**Solution**: I designed a multi-screen interface using the Tkinter library ([2]). Key features include:

- Input fields for job description, role, and interview level specifications.

- Adjustable parameters for fine-tuning the speech-to-text engine.

- A dedicated interview screen with "Speak," "Stop," and "End" buttons, incorporating dynamic behavior to reflect the system's current state.

- Careful thread management to ensure seamless synchronization between the user interface and the interview process.

- A mechanism for gracefully terminating the interview and closing the UI, triggered by a specific LLM-generated phrase ("Thank you for your time").

4. **Answer Accuracy Verification**

**Challenge**: Evaluating the factual accuracy of candidate responses in real-time presented a challenge, given the limitations of LLMs in accessing up-to-date information.

**Solution**: To address this, I developed an accuracy verification system utilizing a Retrieval Augmented Generation (RAG) pipeline with the LangChain framework ([3]):

- Candidate answers are broken down into concise search queries.

- Web scraping techniques are employed to gather pertinent information from Google Search results.

- Retrieved information is embedded and stored in a vector database (Chroma) using OpenAI embeddings.

- A second RAG pipeline retrieves contextually relevant information from the vector store, enhancing the LLM's knowledge base.

- The LLM then assesses the answer's accuracy against this enriched information.

- Inspiration for query decomposition and context management drew upon research on enhancing RAG system effectiveness ([4], [5]).

**5. Sentiment Analysis**

Challenge: To gain a more comprehensive understanding of candidate suitability, we sought to analyze emotional cues present in their responses, going beyond textual analysis. Developing a custom sentiment analysis model proved challenging due to limitations in training data.

**Solution**: I integrated the Hume AI platform ([6]), which specializes in audio and video sentiment analysis. Hume provides a rich set of emotional and sentiment-related metrics, offering deeper insights. This data, after summarization by my colleague, is passed to the final evaluation LLM.

**6. Additional Contributions and System Deployment**

Beyond the core components, I contributed to:

- Implementing robust exception handling throughout the system to enhance stability.

- Creating comprehensive documentation to aid in understanding and future development.

- Collaborating on system testing and defining appropriate evaluation strategies for a system dealing with the inherent non-determinism of human language.

- Deployment planning, including dependency management and exploring executable packaging options.

**7. Conclusion**

My work on the post-processing pipeline, user interface design, and overall system integration has been instrumental in developing a functional and insightful interview screening system. By addressing practical challenges with innovative solutions, this project showcases the potential of AI in recruitment and opens avenues for further exploration in this field.

**Bibliography**

[1] OpenAI, "Whisper," GitHub repository, [https://github.com/openai/whisper](https://github.com/openai/whisper)

[2] Python Software Foundation, "Tkinter - Python interface to Tcl/Tk," [https://docs.python.org/3/library/tkinter.html](https://docs.python.org/3/library/tkinter.html)

[3] Harrison Chase, "LangChain," [https://python.langchain.com/v0.2/docs/introduction/](https://python.langchain.com/v0.2/docs/introduction/)

[4] Rodrigo Nogueira, Jimmy Lin, "From Dense Retrieval to Dense Clustering: Improving Search and Recommendation using Contextual Embeddings," arXiv preprint arXiv:2205.10625, 2022.

[5] Patrick Lewis, Ethan Perez et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems, 2020.

[6] Hume AI, "Hume AI Developer Platform," [https://dev.hume.ai/intro](https://dev.hume.ai/intro)