Technical Report: Automatic Speech Recognition (ASR), Large Language Models (LLM), and Text-to-Speech (TTS)

1. Introduction

This report outlines the technologies explored for **Automatic Speech Recognition (ASR), Large Language Models (LLM), and Text-to-Speech (TTS)** in our prototype. It also discusses the choices made, challenges encountered, and limitations of the current implementation.

2. Technologies Explored

2.1 Automatic Speech Recognition (ASR)

- **Technologies Considered:**
  - Web Speech API (Browser-based ASR)
  - OpenAI Speech-to-Text API
  - speechly
- **Final Choice: Web Speech API**
  - It provides built-in speech recognition in modern browsers.
  - No additional API calls or costs.
  - It was recommended.

2.2 Large Language Models (LLM)

- **Technologies Considered:**
  - OpenAI GPT-4
  - Google Gemini API (Generative AI)
- **Final Choice: Google Gemini API**
  - Easy API integration with structured responses.
  - It’s free.

2.3 Text-to-Speech (TTS)

- **Technologies Considered:**
  - Web Speech Synthesis API

    - OpenAI Text-to-Speech API
    - speechly
- **Final Choice: Web Speech Synthesis API**
    - Works directly in modern browsers.
    - No external API dependencies.
    - Since i have already used it for ASR.

3. Challenges Faced

3.1 ASR Challenges

- **Background Noise Sensitivity:** The ASR system sometimes picks up background noise, leading to false transcriptions(such as fan noise or any electronic device's).
- **Limited Offline Support:** The Web Speech API requires an internet connection, restricting offline usage.
- **Different Browser Behaviors:** Speech recognition works differently across browsers (best in Chrome, limited in Brave and Firefox).

3.2 LLM Challenges

- **Response Consistency:** Ensuring responses remain consistent while keeping the interaction engaging.
- **Latency Issues:** Some delays in processing when sending messages to the Gemini API.
- **Token Limitations:** API constraints on input size affect conversation history.

3.3 TTS Challenges

- **Voice Selection:** The Web Speech Synthesis API does not allow fine-grained control over voice quality.
- **Cross-Browser Issues:** Different browsers use different TTS voices, leading to inconsistencies.
- **Lack of Customization:** Limited ability to modify tone, speed, or style dynamically.

4. Limitations of the Prototype

- **Internet Dependency:** The entire system requires an internet connection for ASR, LLM, and TTS.
- **Limited Language Support:** Language is fixed as English for now.
- **Single-Turn Conversations:** The model does not retain long-term memory beyond session history. And can't detect speech while the speech to text function is running.

5. Conclusion

The prototype successfully integrates **speech recognition, large language models, and text-to-speech capabilities** for real-time conversational AI. While the chosen technologies provide a smooth user experience, challenges such as **background noise sensitivity, latency, and internet dependency** remain areas for improvement. Future iterations could explore **custom ASR models, hybrid LLM approaches, and enhanced voice synthesis options** for a more robust system.

7. References

- Web Speech API Documentation: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API , https://webaudio.github.io/web-speech-api/#introduction
- Google Gemini API: https://ai.google.dev/gemini-api/docs
- Web Speech Synthesis API: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API