COMP47700: Speech and Audio

A+ Presentation Delivery Feedback Tool

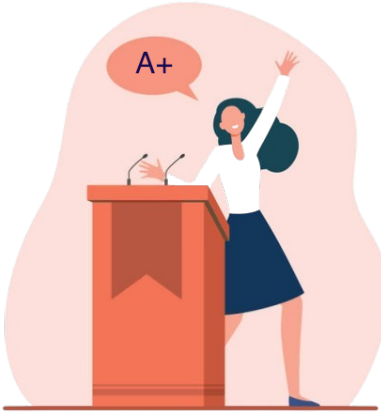
Treasa Murphy

Student ID: 21459632

Table of Contents

Abstract

This project presents a real-time feedback tool designed to help users improve their spoken presentation delivery. By analysing audio input, the system provides metrics such as speech rate, vocal projection, volume, filler word usage, and pause frequency. Implemented using Python and libraries such as librosa, pydub, and speech_recognition, the tool aims to support students and professionals in enhancing their communication skills.



Acknowledgements

Thank you to Dr Alessandro Ragano for his tuition and guidance throughout this module. I'd also like to thank the teaching assistants for their support, and those who tested the tool and completed the survey.

Chapter 1: **Introduction**

Effective presentation delivery is a crucial skill in both academic and professional contexts. However, many individuals encounter challenges such as rapid speech, monotonous tone, excessive use of filler words, and awkward pauses, all of which can adversely affect audience engagement and comprehension.

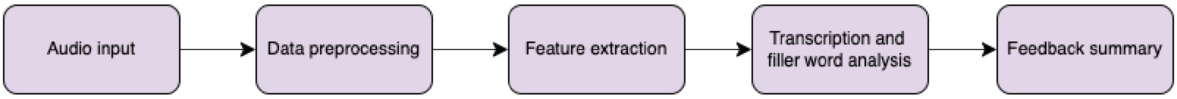
The A+ Presentation Feedback Tool has been developed to address these issues by providing real-time, automated feedback to support the refinement of presentation skills through data-driven insights. Utilising advanced audio processing and speech analysis techniques, the tool evaluates key delivery metrics, including speech rate, vocal inflection, volume consistency, pause frequency, and the occurrence of filler words (e.g. "um", "eh"). These insights are presented through an intuitive Streamlit interface, enabling users to receive immediate and actionable feedback to improve their speaking style.

This project integrates powerful Python-based audio analysis libraries, including librosa, pydub, and speech_recognition, to process both recorded and live speech input. Ultimately, the tool aims to enhance communication proficiency by fostering greater awareness of individual speaking habits.

Chapter 2: System Design

2.1 System Architecture

The tool follows a sequential pipeline structure:



Each component operates independently, facilitating modular development and the potential integration of additional functionalities in future iterations.

2.2 Audio Input and Preprocessing

Users currently provide speech input by uploading a pre-recorded .wav file via the interface. While real-time microphone recording is considered for future development, the current implementation prioritises processing uploaded audio files to maintain a consistent and reliable analysis pipeline.

Upon upload, the audio is standardised using the pydub library, which includes the following preprocessing steps:

- Resampling to a mono channel at a 16 kHz sampling rate, ensuring alignment with common speech processing standards and compatibility with libraries such as librosa and speech_recognition.
- Trimming leading and trailing silence to remove non-speech segments at the start and end of the recording. This adjustment ensures that metrics such as speech rate and pause frequency reflect only the speaker's active delivery.

These preprocessing steps optimise the audio input for subsequent feature extraction, ensuring consistency and enhancing speech-focused analysis.

2.3 Feature Extraction

Following preprocessing, the system extracts a range of quantitative speech features designed to evaluate key aspects of presentation delivery. This process is facilitated using the librosa library, which provides robust tools for analysing time-series audio data.

The extracted features are as follows:

Speech Rate

Estimated as the number of words per minute, calculated using the total word count obtained from transcription and the duration of the audio. This metric identifies whether the speaker's pace is excessively fast or overly slow, both of which can impact listener comprehension.

Pauses

Silent segments within the speech are detected and analysed to determine their frequency, average duration, and distribution throughout the recording. An elevated number of prolonged or irregularly spaced pauses may disrupt the natural flow of speech and affect delivery effectiveness.

Volume Dynamics

The average amplitude and its variance are computed to assess vocal projection and consistency. Limited variation in volume may result in a monotonous delivery, whereas excessive fluctuations can be distracting for the audience.

Inflection (Prosody)

Pitch contour is examined through fundamental frequency (F_0) estimation, enabling an assessment of vocal inflection. This feature helps determine whether the speaker employs expressive intonation or speaks in a flat, monotone manner.

These features collectively provide a multi-dimensional analysis of the speaker's delivery style. By quantifying characteristics that are typically subjectively assessed, the system generates actionable insights that assist users in enhancing clarity, engagement, and overall presentation effectiveness.

2.4 Transcription and Filler Word Detection

To facilitate detailed analysis of spoken delivery, the system integrates an automatic transcription module that converts uploaded speech into text. This functionality is implemented using the speech_recognition Python library, which interfaces with the Google Web Speech API. The API delivers high-quality transcription with minimal latency and supports a diverse range of accents, making it well-suited for varied user demographics.

Once the transcription is generated, it is systematically analysed to identify the presence of filler words, such as "um" and "eh" , along with other disfluencies. These terms are detected using pattern matching techniques that scan the transcribed text for predefined filler expressions. The system then calculates the frequency and proportion of filler words relative to the total word count, providing users with quantitative feedback on their speech fluency.

Detecting filler words is crucial for assessing verbal fluency, as excessive use can diminish a speaker's credibility and divert attention from the core message. By highlighting these tendencies, the tool promotes greater awareness of speech patterns, encouraging more deliberate and confident delivery.

While the current implementation relies on predefined word lists, future iterations may incorporate natural language processing (NLP) techniques to improve detection accuracy by identifying hesitations, repetitions, and context-specific disfluencies with greater nuance.

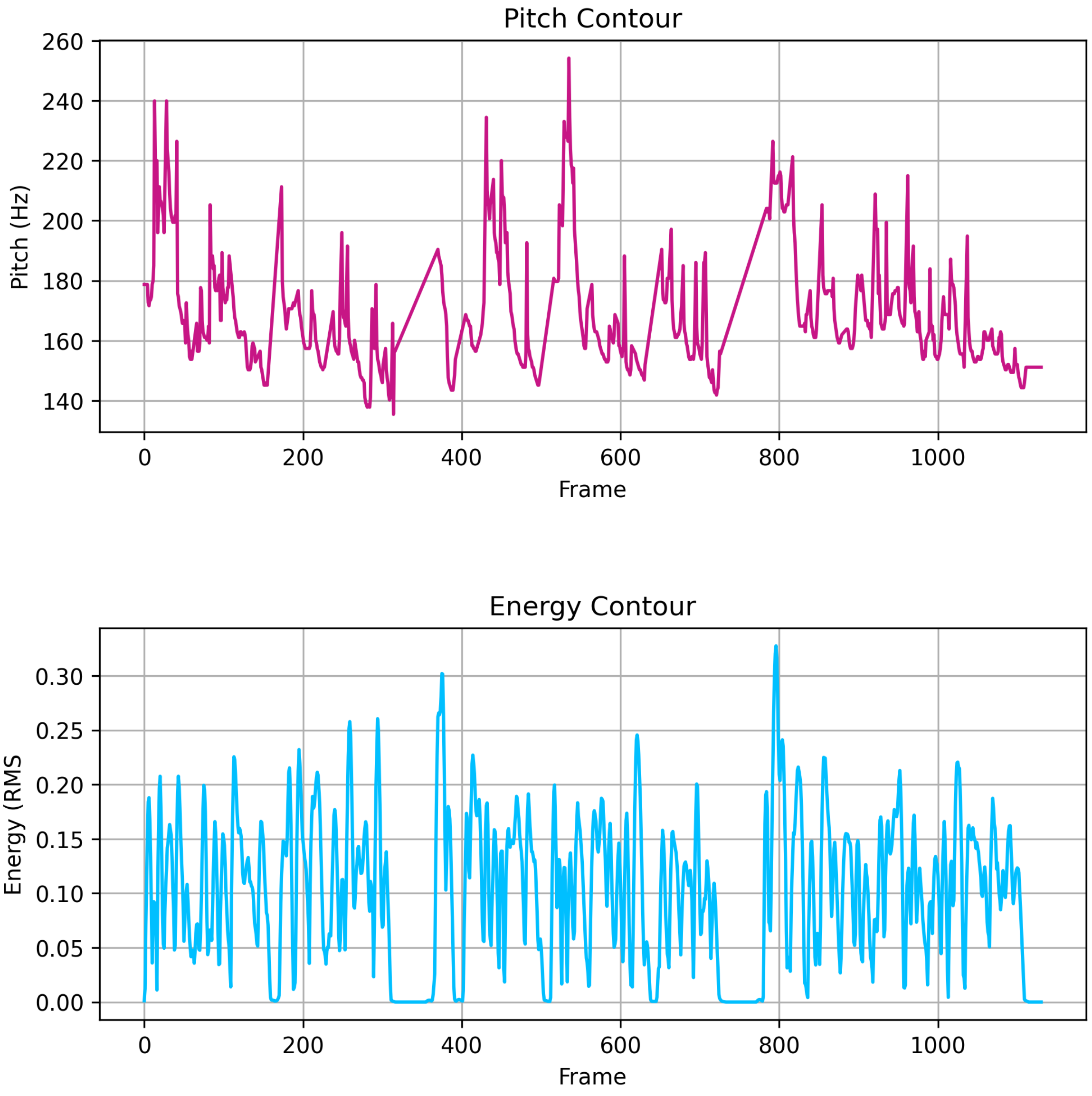
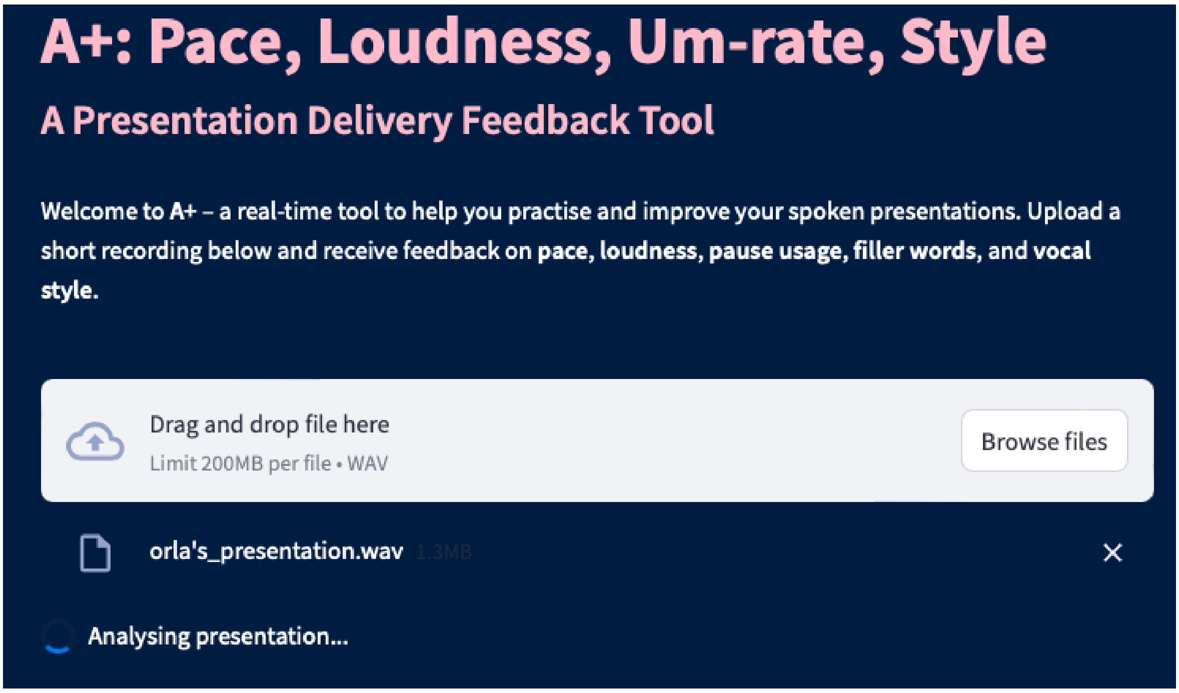


Figure 2.1: Visual feedback generated by the tool: vocal energy (top) and pitch contour (bottom), used to assess projection consistency and vocal inflection.

2.5 Streamlit Interface

The user-facing component of the A+ Presentation Feedback Tool is developed using Streamlit, a lightweight Python framework designed for building interactive data applications. The interface prioritises intuitiveness and accessibility, allowing users to engage with the tool without requiring technical expertise.



A+: Pace, Loudness, Um-rate, Style

A Presentation Delivery Feedback Tool

Welcome to A+ – a real-time tool to help you practise and improve your spoken presentations. Upload a short recording below and receive feedback on **pace, loudness, pause usage, filler words, and vocal style.**

Drag and drop file here

Limit 200MB per file • WAV

orla's_presentation.wav 1.3MB

Analysing presentation...

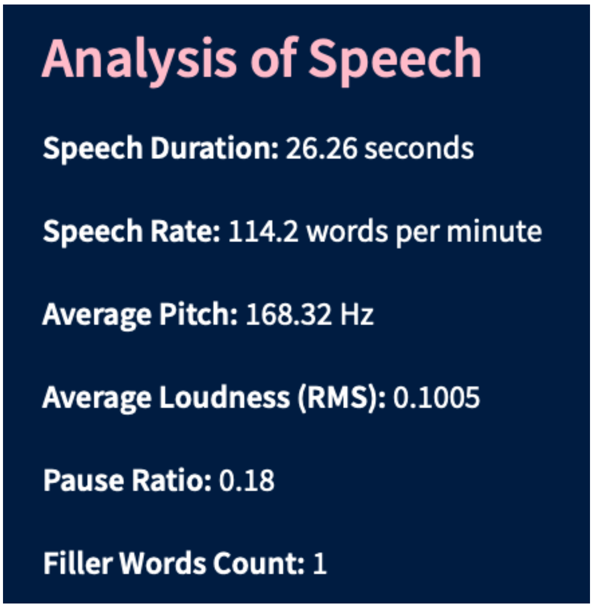
Figure 2.2: Streamlit interface of the A+ Presentation Feedback Tool, displaying delivery metrics and visual feedback.

Upon accessing the interface, users are prompted to upload a .wav audio file containing their spoken presentation, as shown in Figure 2.2. Once processed, the system dynamically generates a set of delivery metrics and visualisations that summarise the speaker's performance.

2.5.1 Key Elements of the Streamlit Interface

Summary Metrics

As outlined in Figure 2.3, the tool delivers structured quantitative insights on speech rate, average pause rate, average loudness, and filler word usage. These metrics provide a precise, data-driven assessment of delivery quality. Users with a background in speech analysis or audio processing can interpret these numerical values within the broader context of acoustic science, allowing them to derive technical insights on pacing, projection, and fluency.



Analysis of Speech

Speech Duration: 26.26 seconds

Speech Rate: 114.2 words per minute

Average Pitch: 168.32 Hz

Average Loudness (RMS): 0.1005

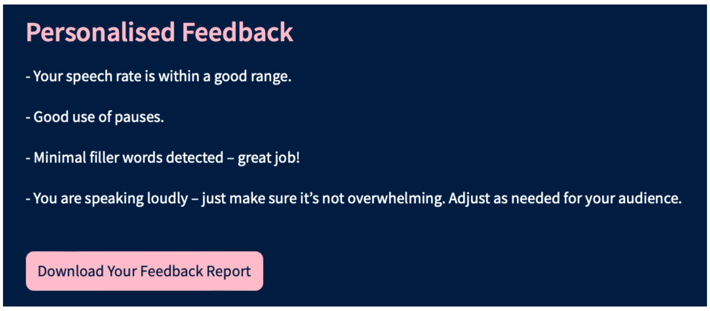
Pause Ratio: 0.18

Filler Words Count: 1

Figure 2.3: Structured feedback panel displaying quantitative metrics for speech rate, pause duration and frequency, volume variation, and filler word usage. These values provide a concise assessment of delivery quality.

Textual Feedback

To ensure accessibility for non-technical users, brief descriptive insights accompany the extracted metrics, as demonstrated in Figure 2.4. These explanations translate complex acoustic features into practical recommendations. For instance, if a high frequency of filler words is detected, the user may receive a message such as "Consider reducing filler words to improve fluency." This structured feedback simplifies otherwise intricate speech characteristics, making it easier for presenters to understand and act upon their strengths and areas for improvement.



Personalised Feedback

- Your speech rate is within a good range.

- Good use of pauses.

- Minimal filler words detected – great job!

- You are speaking loudly – just make sure it's not overwhelming. Adjust as needed for your audience.

Figure 2.4: Personalised textual feedback based on extracted delivery metrics, offering user-friendly suggestions such as reducing filler words or adjusting speech rate.

The interface is designed to facilitate iterative practice, enabling users to record, receive feedback, and refine their delivery over multiple sessions. While the current functionality is limited to .wav file uploads, future iterations may integrate live audio recording and personalised feedback suggestions based on historical performance. By combining structured metrics for technical users with accessible feedback for general presenters, the tool bridges the gap between speech science and practical communication improvement.

Chapter 3: Implementation

The A+ Presentation Feedback Tool is implemented in Python, utilising a combination of open-source libraries for audio processing, speech analysis, and interface development. The system is designed with a modular architecture, comprising distinct components responsible for input handling, feature extraction, transcription, and feedback presentation.

3.1 Core Libraries

The following Python libraries are employed to support various functionalities:

- pydub: Handles audio pre-processing, including format conversion, resampling to 16 kHz, mono channel enforcement, and silence trimming.
- librosa: Extracts acoustic features such as pitch (fundamental frequency), amplitude, and timing information relevant to speech dynamics.
- speech_recognition: Facilitates automatic transcription via the Google Web Speech API, enabling downstream analysis of speech content and filler word detection.
- matplotlib: Generate visual feedback, including bar charts and plots, to present extracted speech features.
- Streamlit: Provides the web-based interface, allowing users to upload .wav files and view interactive feedback in real time.

3.2 System Workflow

The main application logic is encapsulated within modular functions to ensure scalability and maintainability. Upon file upload, the audio is processed and stored locally, after which feature extraction functions analyse the waveform to compute key delivery metrics, including:

- Speech rate (words per minute)
- Number and duration of pauses
- Average volume
- Pitch variability

The transcription module processes the speech data separately, identifying filler word occurrences using a predefined list.

The extracted results are compiled into a set of visual and textual outputs, which are displayed via the Streamlit interface. The tool is lightweight and designed for ease of use, requiring only Python and a modern web browser to operate efficiently.

Chapter 4: Evaluation

The A+ Presentation Feedback Tool was evaluated through a combination of functional testing and user feedback to assess its accuracy, stability, and overall effectiveness.

4.1 Functional Testing

Initial testing focused on verifying the reliability of key system components, including audio preprocessing, feature extraction, transcription, and interface rendering. A diverse set of .wav files containing variations in speech patterns, noise levels, and delivery styles was used to evaluate the tool's robustness under different conditions.

4.2 User Feedback

To assess usability and practical effectiveness, a Google Forms survey was distributed to a small group of volunteers who used the tool to analyse their own presentation recordings. The survey gathered both quantitative ratings and qualitative feedback on aspects such as interface usability, clarity of feedback, and perceived usefulness. Participants were primarily students from varied academic backgrounds.

4.3 Findings and Insights

Overall, user feedback was predominantly positive, with participants highlighting the ease of use and the value of receiving objective metrics on their speaking habits. Features such as speech rate calculation and filler word detection were particularly appreciated for their role in identifying areas for improvement.

Several users suggested potential enhancements, including:

- Providing resources on improving presentation skills, such as videos on reducing filler words.
- A tracker to record how many times the tool is used in preparation for a presentation.
- Indicating sections where a speaker is too soft.
- Support for multiple languages to increase accessibility.
- A real-time recording widget to allow instant feedback.
- Making it easier to add audio recordings.

- Comparative analysis of multiple presentation recordings.
- Providing sample presentations demonstrating effective delivery techniques.
- Offering language suitability assessments or suggesting synonyms for repeated words.

These suggestions provide valuable insights into potential directions for enhancing the tool's functionality and user experience.

4.4 Conclusion

Although the sample size was limited, the evaluation provided valuable insights into the tool's effectiveness and usability. User feedback directly informed priorities for future development, ensuring continued refinement. The results confirm that the tool successfully meets its core objective of delivering accessible, data-driven feedback to support improved spoken presentation delivery.

Chapter 5: Future Work

While the A+ Presentation Feedback Tool establishes a solid framework for analysing spoken presentation delivery, several avenues for future development could significantly enhance both its functionality and user experience.

5.1 Real-Time Speech Recording

Integrating a real-time recording widget within the Streamlit interface would eliminate the need for users to manually upload .wav files. Instead, users could record speech directly within the browser. This feature would streamline the workflow and facilitate more immediate feedback, improving the overall user experience. Additionally, enabling real-time recording would allow users to compare multiple presentation recordings as they refine their delivery.

5.2 Normative Benchmarks for Delivery Metrics

Leveraging a training corpus, such as transcripts and audio from TED Talks, could establish normative benchmarks for key delivery metrics. Examples of these metrics include optimal speech rate and pitch variation. By comparing users' speech characteristics to established patterns of effective public speakers, the tool could provide more personalised and comparative feedback, offering insights beyond raw metrics.

5.3 Real-Time Feedback on Delivery Quality

Extending the tool's capabilities to offer real-time feedback is another long-term objective. By pinpointing moments in the recording where the speaker's pace fluctuates, whether excessively fast, overly slow or optimally modulated, the system could generate a timeline of delivery quality. This would help users identify sections of their presentation that require further refinement. Additionally, the tool could highlight sections where speakers are too soft, providing targeted suggestions for improving clarity and volume.

5.4 Additional Enhancements

Future iterations of the tool may incorporate the following features:

- Transcript highlighting, where filler words and disfluencies are visually marked within the transcribed text to improve user awareness.
- Support for multiple languages and accents, enhancing accessibility for non-native English speakers and enabling its use across diverse linguistic backgrounds.
- Machine learning integration to score presentations holistically or classify delivery styles, such as confident or hesitant.
- Gamification elements, such as progress tracking and speaker badges, to encourage continuous practice and improvement.
- A tracker for recording tool usage, helping users monitor how many times they use the tool in preparation for their presentation.
- Providing resources on improving presentation delivery, such as video samples demonstrating effective speaking techniques and advice on avoiding filler words.
- Synonym suggestions for repeated words, ensuring users maintain varied and engaging speech throughout their presentation.
- Expanding real-time functionality to offer instant insights on pacing, pitch variation and overall speech clarity.
- Tailoring feedback for specific purposes, including targeted advice for practising job interviews or professional presentations.

These extensions would transform the tool from a static feedback system into a more interactive, adaptive and pedagogically valuable resource for improving spoken communication.

Chapter 6: **Conclusion**

The A+ Presentation Feedback Tool was developed to assist users in improving spoken presentation delivery through accessible and data-driven feedback. By integrating audio pre-processing, acoustic feature extraction, automatic transcription, and an interactive Streamlit interface, the tool provides detailed insights into key delivery metrics, including speech rate, pauses, vocal projection, and filler word usage.

Initial testing and user feedback confirm that the tool is functional, effective, and particularly valuable for students and professionals seeking to enhance confidence and clarity in their speech. Users highlighted the value of objective metrics and suggested additional features, such as real-time recording, comparative analysis of multiple presentation recordings, and language adaptability. These insights underscore the potential of the tool to expand its scope to support various presentation contexts, including job interview preparation and multilingual accessibility.

Although the current implementation is centered on.wav file uploads and fundamental metrics, the underlying architecture is designed to accommodate future enhancements. Planned developments include dynamic feedback mechanisms, real-time speech recording, expanded linguistic support, and machine learning-based scoring to provide personalised insights into delivery styles.

The project underscores the importance of combining speech analysis with visual feedback to facilitate more effective communication. The tool establishes a strong foundation for ongoing development and positions itself as an adaptive and pedagogically valuable resource to improve spoken presentation skills.

Bibliography

1. Huang, M., Bahmanyar, S. & Wiggins, J. Voice Coach: Real-Time Feedback for Oral Presentation Skills Using Speech Analysis. *Proceedings of the ACM on Human-Computer Interaction* **6,** 1–26. https://dl.acm.org/doi/fullHtml/10.1145/3491101.3519611 (2022).
2. MacPherson, M. K. & Smith, A. Speech Rate and Communicative Effectiveness in Individuals with Parkinson's Disease. *Frontiers in Human Neuroscience* **13,** 132. https://pmc.ncbi.nlm.nih.gov/articles/PMC6505544/ (2019).
3. Narakeet. *Text-to-WAV Audio Converter* https://www.narakeet.com/create/text-to-wav.html. 2025.
4. Bernstein, L. *The trick to powerful public speaking* https://www.ted.com/talks/lawrence_bernstein_the_trick_to_powerful_public_speaking. 2019.
5. Cafaro, A., Vilhjálmsson, H. H. & Bickmore, T. Exploring Feedback Strategies to Improve Public Speaking: An Interactive Virtual Audience Framework. *ResearchGate.* Accessed May 3, 2025. https://www.researchgate.net/publication/292148167_Exploring_feedback_strategies_to_improve_public_speaking_an_interactive_virtual_audience_framework (2016).

Chapter 7: Appendix

The source code for the *A+ Presentation Feedback Tool* is publicly available on GitHub:

https://github.com/treasa-murphy/a-plus-presentation-feedback-tool

Prerequisites

Before installing the Python dependencies, make sure `ffmpeg` is installed on your system.

To install `ffmpeg` on macOS, run:

```
brew install ffmpeg
```

On Windows, you can download and install it from the official FFmpeg website: https://ffmpeg.org/

You will also need Python 3.9 or higher installed.

To run the tool locally:

1. Clone the repository:

   ```
   git clone https://github.com/treasa-murphy/a-plus-presentation-feedback-tool.git
   cd a-plus-presentation-feedback-tool
   ```

2. Create a virtual environment and install dependencies:

   On macOS/Linux:

   ```
   python -m venv venv
   source venv/bin/activate
   pip install -r requirements.txt
   ```

   On Windows:

   ```
   python3 -m venv venv
   venv\Scripts\activate
   pip install -r requirements.txt
   ```

3. Launch the Streamlit interface:

   ```
   streamlit run app.py
   ```

4. Upload a `.wav` file when prompted to receive delivery feedback. Sample `.wav` files are included in the project directory for testing purposes.