Automation of Sound Quality Assessment for Voice Calls

Proposal - 2023

Automation proposal

- Divide entire system into modular pieces that can each be automated independently of each other
  - Call generation and audio recording
  - Voice quality analysis
- Modularity can also allow for different hardware setups to be used in each component
- Elimination of as many variables as possible that can affect the audio quality to create the control setup

Automation of Call Generation and Audio Recording

- Currently, calls are done fully manually using microphone setups and the phone loudspeaker, connected to an audio interface and an audio server running QjackCtl
- This can be expanded upon and used to automate phone calls and increase the quality of the audio recording, to improve the accuracy of the voice analysis.
    - Various test audio can be recorded/generated on test server
    - Audio can be piped to the calling phone from the computer as the call is made to the receiving phone
    - Receiving phone can also be connected directly to the audio interface to record the audio from the phone output
    - Calls can be made individually or automated and batched

Automation of Call Generation and Audio Recording

- Calling phone:
  - TRRS Splitter
  - Analog Cables
  - Microphone
  - Sound Mixer
- Receiving Phone:
  - TRRS Splitter
  - Analog Cables
  - DAC/Audio Interface
- Audio Server:
  - QjackCtl
  - SIP Server
  - Connections to both calling phone and receiving phone

Automation of Voice Quality Analysis

- Current stated target metric is PESQ with full reference (FR) analysis
- POLQA has been adopted as the successor standard to PESQ, requires license to use
- POLQA open source implementations are available in limited form in a few Python libraries, however the consistency has not been tested

Automation of Voice Quality Analysis

- Virtual Speech Quality Objective Listener (ViSQOL): fully open-source alternative to POLQA
- General full-reference objective speech quality metric with a particular focus on VoIP degradations
- Feature selection:
  - Time alignment via segmentation of the sample and application of the Neurogram Similarity Index Measure (NMIS)
  - Predicting Warp via NMIS comparisons between test and reference patches
  - Similarity comparison using the structural similarity index (SSIM). Treat spectrograms as images and compare directly with the SSIM via pixel comparisons
    - Intensity -> luminance
    - Variance -> contrast
    - Cross-correlation -> structure
  - Sigmoid mapping function to translate NSIM similarity score onto a MOS-LQOn score for scoring speech quality

Automation of Voice Quality Analysis

- Pre-processing of the audio signal will need to be done in accordance with the respective pre-processing guidelines outlined in the ViSQOL algorithm/standards.
- Usually this will involve scaling degraded signals to match the power levels of the reference signal, then applying FFTs across recommended windows.
- Preprocessing steps:
  - audio downsampling
  - frame alignment
  - time-warping detection
  - sample/patch window determination
  - FFT
- VISQOL API: https://github.com/google/visqol

Automation of Voice Quality Analysis

- Alternatives to ViSQOL:
  - Audio Quality Pipeline for Python: https://github.com/JackGeraghty/AQP
  - Contains both the WARP-Q and PESQ speech quality metrics

Timeline

- Phased Project Schedule
  - Phase 1: Exploratory/prototyping
  - Phase 2: System testing and fine-tuning
    - Offsite device testing
    - In-lab device testing

Phase 1 Prototyping

- Period: 3 Months
- 1st month: exploratory phase
  - Use various audio hardware setups as well as implementations of the voice quality algorithms to test
    - Efficacy of automation
    - Time/labor/complexity cost of setup
    - Efficacy of the voice quality algorithms
  - Start to finalize prototype before next phase
- 2 months: fine-tuning and testing
  - Finalize setup
  - Automation of call generation and batch processing of audio