TableTalk Human-AI Technical Test

Narrative Voice Processing and Classification

Overview

The TableTalk project aims to build a system that organizes and retrieves narrative voice recordings used in interactive storytelling environments such as tabletop role-playing games. These recordings may include narration, character dialogue, environmental descriptions, or dramatic moments that are triggered during gameplay.

The goal of this test is to evaluate your ability to process voice recordings, apply machine learning techniques to speech data, and develop tools that support classification and retrieval of narrative audio.

Because the TableTalk dataset is still being developed, this test uses publicly available speech datasets to simulate the kinds of tasks required by the system.

Applicants will complete several tasks involving audio preprocessing, classification of narrative tone, AI-based transcription, and prototype audio retrieval.

Dataset

Applicants may use any publicly available speech dataset.

Recommended options include:

RAVDESS Emotional Speech Dataset
https://zenodo.org/record/1188976

CREMA-D Emotional Speech Dataset
https://github.com/CheyneyComputerScience/CREMA-D

Mozilla Common Voice Dataset
https://commonvoice.mozilla.org

Applicants may work with a subset of the dataset (for example **50–200 audio recordings**).

Task 1: Audio Processing Pipeline

Create a Python script that processes a collection of speech recordings.

Your script should:

• load audio files
• normalize or standardize audio levels
• segment audio if necessary
• extract useful audio features

Examples of useful features include:

• MFCC coefficients
• pitch
• spectral centroid
• energy
• duration

The goal is to convert the raw audio files into a structured dataset suitable for machine learning.

Deliverables

Provide:

• Python code for audio processing
• description of extracted features
• example output dataset

Task 2: Narrative Tone Classification

TableTalk recordings will eventually include different **narrative tones**, such as:

• suspense
• calm description
• urgency
• dramatic emphasis
• character dialogue

Using your chosen dataset, train a model that classifies recordings based on **emotional tone or speech characteristics**.

Possible approaches include:

• logistic regression
• random forest
• neural networks
• pretrained audio embeddings

If your dataset already includes emotional labels, you may use them. Otherwise, you may define your own categories based on audio characteristics.

Deliverables

Provide:

• code for the classification model
• description of the approach
• evaluation metrics (accuracy, F1 score, etc.)
• short discussion of results

Task 3: AI-Based Transcription

Implement an automatic speech transcription pipeline.

You may use any available speech-to-text model such as:

• OpenAI Whisper
• Vosk
• HuggingFace speech models
• DeepSpeech

Your program should:

• transcribe multiple audio recordings
• output transcripts in text format
• measure transcription accuracy on a small subset

Deliverables

Provide:

• transcription code
• example transcripts
• short discussion of transcription quality

Task 4: Narrative Audio Retrieval (TableTalk Simulation)

Design a simple prototype system that retrieves voice recordings based on narrative characteristics.

For example, the system might support queries such as:

• “calm narration longer than 4 seconds”
• “high-energy speech”
• “dramatic dialogue”

The system does not need to be a full application. A simple script that filters or retrieves matching recordings is sufficient.

Deliverables

Provide:

• explanation of retrieval method
• example search queries
• results returned by the system

Storytelling Audio Analysis

TableTalk recordings will involve storytelling narration rather than simple speech.

For this bonus task, select several recordings and analyze features that might distinguish **storytelling narration from conversational speech**.

Examples of features may include:

• pacing and pauses
• pitch variation
• energy dynamics
• sentence length

Discuss which features might help identify storytelling narration automatically.

Submission Requirements

Your submission should include:

1. Source code (preferably a GitHub repository)
2. Short technical report (2–4 pages) explaining your approach
3. Instructions for running the code
4. Example outputs from your system

Evaluation Criteria

Submissions will be evaluated based on:

- correctness of implementation
- clarity and structure of code
- ability to process audio datasets
- quality of machine learning approach
- clarity of written explanation
- creativity in approaching narrative audio tasks

Estimated Effort

Expected time to complete the test: **6–10 hours**

Recommended Tools

Python libraries commonly used for speech processing:

- librosa
- numpy
- pandas
- scikit-learn
- pytorch / torchaudio
- transformers
- whisper