York to New York: Interpolating Accents in Text To Speech Synthesis

Richard Ahn (rahn) & Shawn Krishnan (shawnakk) & Benjamin Lash (balash)
10-423/623 Generative AI Course Project

May 2, 2025

1 Introduction

Text-to-speech (TTS) systems have diverse applications, yet their effectiveness significantly varies based on several usability factors, including the quality of generated audio, realism of synthesized voices, and the capacity to convey emotional nuances such as empathy. Motivated by the goal of enhancing realism in AI-generated voices, we focus specifically on accents, which substantially contribute to perceived naturalness and relatability. Our proposed method aims to surpass existing models by accurately synthesizing voices with blended accents, reflecting the linguistic diversity found in today's increasingly global population. Specifically, we introduce a model that accepts a distribution across multiple accent classes as input, processes this vector through a feed-forward neural network, and outputs a continuous accent embedding used to condition the TTS system. The effectiveness of our approach is evaluated using a speech-to-text system, benchmarked against a state-of-the-art accent detection algorithm to measure the accuracy and realism of generated speech samples, and assessed by participants in a survey.

2 Dataset / Task

For both the Amazon and mixed-accent systems we now train on Common Voice 12.0 Ardila et al. [2020] Because many clips lack explicit accent labels, we infer a label for every training utterance by running the CommonAccent classifier [Zuluaga-Gomez et al., 2023]. When a discrete accent label is necessar, we take the *argmax* of its predicted distribution. For the Grad-TTS baseline we continue to use the LJ Speech corpus Ito and Johnson [2017]. All audio is resampled to 16 kHz.

Our primary task is to generate speech conditioned on a user-defined accent distribution. Throughout this work we focus on two target accents (Indian English and American English) and a $50 : 50$ blend of the two although our model is trained to handle any distribution over a total of 16 accents. Our model extends GradTTS and and Amazon's diffusion-based TTS framework, by introducing flexible accent conditioning.

To evaluate the synthesized speech, we will utilize three metrics:

Accent Fidelity We will employ a state-of-the-art accent detection system to determine how accurately our synthesized speech reflects the intended accent distribution. Evaluations include comparisons between our reimplementation of Amazon's TTS model and our proposed mixed-accent system. We measure top-1 accuracy and mean squared error (MSE) between input accent distributions and those predicted by the detection system.

We also distributed a survey asking participants to classify samples generated by our model according to accent. This survey was distributed through Piazza, social media, and personal networks.

Text Accuracy Generated speech is evaluated using a speech-to-text system to quantify how precisely synthesized audio matches the original textual prompts. This measure will serve as an indicator of intelligibility and accuracy.

Perceptual Realism To further assess realism, we include questions about audio quality in the aforementioned survey. We ask participants to rate the quality of the audio samples and calculate a Mean Opinion Score (MOS).

3 Related Work

Recent advancements in text-to-speech (TTS) synthesis have greatly improved the modeling of accents and speaker characteristics. Our research builds upon several influential prior studies summarized below.

Diffusion-Based Accent Modeling in Speech Synthesis. Deja et al. [2023] proposed a diffusion-based model specifically tailored to accent synthesis. Their method effectively captures subtle differences among several English accents and has demonstrated improved performance over previous approaches. Furthermore, they introduced a saliency-map-based accent conversion technique within the diffusion framework to facilitate transformation between accents.

**Grad-TTS: A Diffusion Probabilistic Model for

Text-to-Speech. Popov et al. [2021] developed Grad-TTS, which employs a diffusion probabilistic approach for generating mel-spectrograms through incremental denoising. The method utilizes stochastic differential equations, offering a good balance between audio quality and generation speed, achieving results competitive with contemporary TTS systems as measured by Mean Opinion Scores (MOS).

Controllable Accented Text-to-Speech Synthesis with Fine- and Coarse-Grained Intensity Rendering. Liu et al. [2022] introduced a neural TTS architecture enabling nuanced control of accent intensity. Their approach features an accent variance adaptor that explicitly models accent-specific variations in pitch, energy, and duration. Notably, this model was primarily trained on Mandarin-speaking non-native accents, controlling accent strength through a scalar intensity parameter.

Accent Recognition with Hybrid Phonetic Features. Zhang and Chen [2021] designed an accent recognition model that employs hybrid phonetic features derived from an auxiliary automatic speech recognition (ASR) task. Their framework integrates acoustic embeddings from both fixed and trainable representations, enhancing robustness and accuracy for accent classification tasks.

Multi-Scale Accent Modeling and Disentangling for Multi-Speaker Multi-Accent Text-to-Speech Synthesis. Zhou et al. [2025] introduced a multi-scale accent modeling framework for handling multiple speakers and accents in TTS. Their method captures both global utterance-level and local phoneme-level accent variations, enabling disentangled control over speaker identity and accent characteristics. This significantly enhances flexibility and naturalness in multi-accent synthesis.

Accent Conversion in Text-to-Speech Using Multi-Level VAE and Adversarial Training. Melechovsky et al. [2024a] proposed a TTS model employing a multi-level variational autoencoder (VAE) combined with adversarial training to enhance accent conversion. Their method models accent-specific variations effectively and improves conversion quality compared to baseline methods, advancing inclusive speech technology.

AccentBox: Towards High-Fidelity Zero-Shot Accent Generation. Zhong et al. [2025] developed AccentBox, a two-stage pipeline for high-fidelity zero-shot accent synthesis. It uses a robust accent identification model to extract speaker-independent accent embeddings, which then condition a zero-shot TTS system, enabling realistic accent generation even for unseen accents and speakers.

DART: Disentanglement of Accent and Speaker Representation in Multispeaker Text-to-Speech. Melechovsky et al. [2024b] introduced DART, a method that disentangles speaker and accent characteristics using multi-level VAEs and vector quantization. This enables precise, independent control over speaker identity and accent attributes in multispeaker TTS systems.

Our project builds on these prior contributions by conditioning the TTS model on blended distributions of native English dialects, rather than exclusively on non-native speaker accents. Moreover, we intend to leverage existing robust accent recognition approaches, such as the model developed by Zhang and Chen [2021], to objectively assess the accuracy of our synthesized accent distributions.

4 Approach

In this work, we propose a modification of Amazon's text-to-speech framework, which is based on the Grad-TTS model. Grad-TTS uses a diffusion model that works in the Mel Spectrogram space. In Grad-TTS training, the model is prompted with a text sample from the training data and the output audio is compared against the corresponding audio sample in the training dataset to get the loss. Amazon extends this by also conditioning over an accent embedding as determined by the accent label in the training dataset. In our mixed accent model, generation is conditioned over multiple accents rather than a single one. Our method uses a distribution across accent classes as input, which is mapped through a feedforward network to produce a continuous accent embedding. This embedding network is trained jointly with the rest of the model.

During training, the accent distribution is derived from a pre-trained accent classification network applied to each speech sample. At inference time, users can directly specify the desired distribution, allowing for controllable mixed-accent synthesis. Our baseline method is the original Grad-TTS implementation. Our aim is to replicate its core speech quality metrics while extending its functionality. The key contribution of our project is a novel generative model that enables flexible accent blending in speech.

For the Amazon-style model we convert each one-hot accent ID to an embedding. During training, the accent id is obtained by *argmaxing* the accent classifier output to obtain a single class label. For the mixed-accent model we feed the entire accent distribution vector into a two-layer feed-forward network to obtain a continuous accent embedding. During training, we obtain this accent distribution vector from the probability distribution produced by the accent classification network. Outside of training, the distirbution is provided by the user.

5 Experiments

To fully evaluate our proposed approach, we conduct experiments assessing the precision, accent fidelity, and perceptual naturalness of synthesized speech.

All inference results in this section use **100 diffusion steps**.

5.1 Accent Fidelity Experiment

We quantify how accurately our generated speech samples reflect the intended input accent distributions. Specifically, we employ an existing robust accent detection system, CommonAccent (not to be confused with the CommonAccent dataset on which it is trained) Zuluaga-Gomez et al. [2023], to classify synthesized audio and produce accent probability distributions.

We evaluate three systems:

1. Our reimplementation of Amazon's accent-conditioned TTS system.

2. Our mixed-accent TTS system conditioned on a 1-hot vector representing a single accent.

3. Our mixed-accent system conditioned on an accent spread (i.e., a probability distribution).

For each of these systems, we generate audio samples and assess their top-1 accuracy using the classifier.

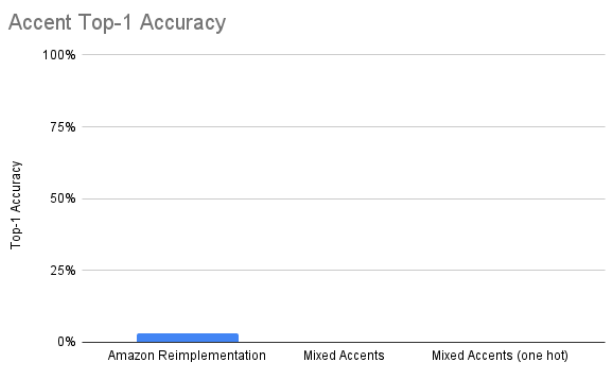

Figure 1: Top-1 accuracy of accent detection

We also assess mean squared error (MSE). For each sample, we compare either the 1-hot accent vector (Amazon TTS system) or the probability distribution (our system) to the predicted distribution from the classifier.

In the end, our results were extremely poor, with near zero accuracy. In all of our experiments, the accent classifier classified over 50% of synthetic audio samples as American-accented. This is likely due to the poor audio quality. For this reason, we added a question about accent discernment to our survey discussed later.

Running the accent fidelity experiments took about one hour of computation in total.

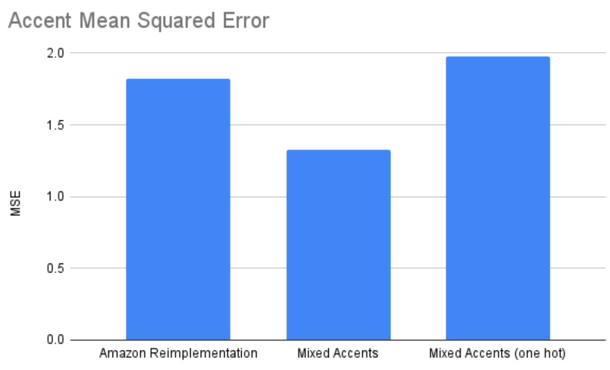

Figure 2: Mean squared error of accent distribution

5.2 Speech-to-Text Accuracy Experiment

To evaluate intelligibility and textual fidelity, we transcribe synthesized speech using the Google Cloud Speech-to-Text API. The transcription is then compared against the original input text to calculate accuracy. We chose 100 sentences randomly sampled from the LJ Speech dataset. The sentences had an average length of 16.89.

We calculate the percentage of clips where the Google-assessed text matches the output. We also calculate average Levenshtein distance from the input text to the transcription. Levenshtein distance measures the number of deletions, insertions, and replacements necessary to go from one sequence to another.

Running the speech-to-text accuracy experiment took about an hour of computation time in total across all models.

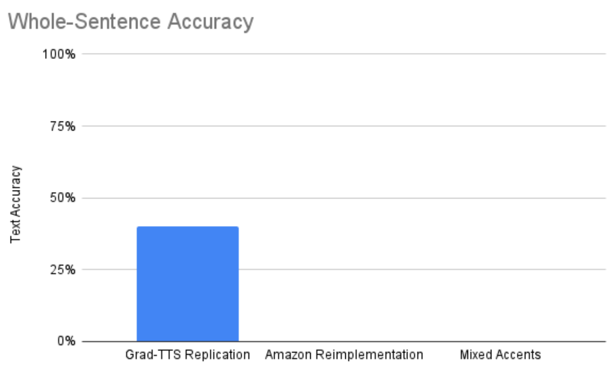

Figure 3: Text accuracy of speech-to-text transcription

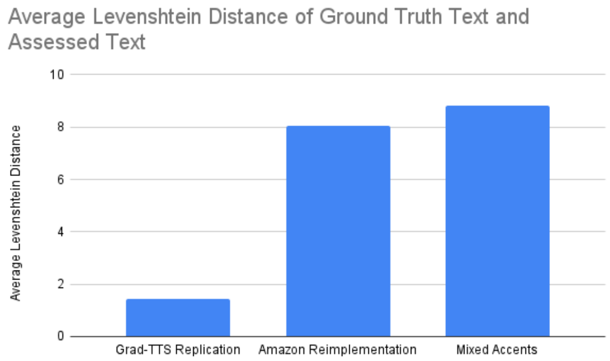

Figure 4: Average Levenshtein distance from ground truth text to speech-to-text transcription. The average prompt length is 16.87 words.

5.3 Human Accent-Identification Survey

We designed a concise survey in which participants classify the accent of generated speech rather than decide whether it is human-made (as was in our original proposal).

We drew three sentences from the Universal Declaration of Human Rights. For each sentence, we synthesized three audio clips with our mixed accent model and presented them to the user:

1. An Indian-accented clip,
2. An American-accented clip,
3. A 50–50 blended Indian-American clip, and

For each sentence, respondents answer the following shuffled questions:

- Which clip is **Indian**-accented?
- Which clip is **American**-accented?
- Which clip sounds **in-between**?

For each sentence we also presented an Indian-accented clip and an American-accented clip generated by our implementation of Deja et al. [2023] and asked participants which was Indian-accented and which one was American-accented.

We found that users generally classified the samples produced by the mixed accent model correctly but generally incorrectly classified samples produced by the reimlpementation of Deja et al. [2023]. Users correctly classified 64.4% of samples from the mixed accent model and only 45.10% of samples from the Amazon reimplementation.

After completing the accent classification task, participants evaluated audio quality. For **five** separate text prompts, we generate samples using each of the three models (Grad-TTS baseline, Amazon reimplementation, and mixed-accent). The resulting 15 clips are shuffled, and listeners rate the quality of each on a scale from 1 to 5 with 0.5-point increments. Because we did not use paid crowd-workers, we reduced the number of samples compared to the 40-clip evaluation in Popov et al. [2021]. From this we calculated Mean Opinon Score (MOS) for each model and compared against our baseline, Popov et al. [2021].

Our survey was answered by 22 people.

Model	MOS with 95% confidence interval
Grad-TTS-1000 (baseline)	4.44 ± 0.05
Grad-TTS-100 (baseline)	4.38 ± 0.06
Grad-TTS-10 (baseline)	4.38 ± 0.06
Grad-TTS-4 (baseline)	3.96 ± 0.07
Grad-TTS Replication	4.42 ± 0.14
Amazon Replication	2.33 ± 0.17
Mixed Accent Model	1.96 ± 0.17

Table 1: Mean Opinion Score by Model

A detailed description of the survey structure appears in Appendix A.

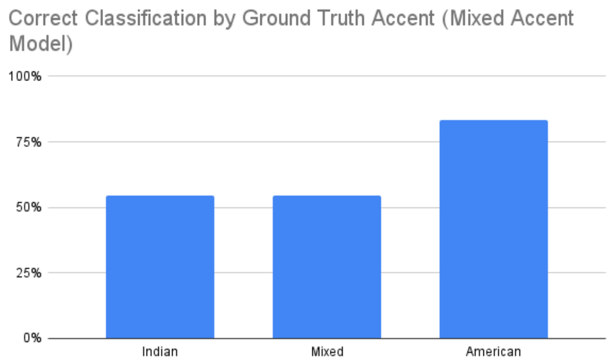

Figure 5: Average percentage of correct accent classifications by survey participants by ground truth accent (Mixed Accent)

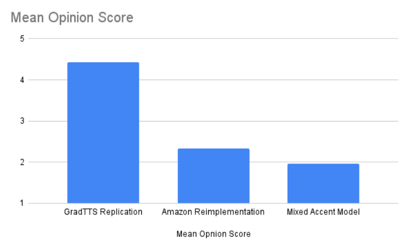

Figure 6: Mean Opinion Scores (MOS)

6 Code Overview

Our implementation is based on a modification of the Grad-TTS repository. In our fork of this repository, which we called MixedAccentTTS, we introduced custom modules for accent blending, distribution conditioning, and evaluation. We created a main branch for the Grad-TTS replication, an Amazon-Reimplementation branch for reimplementing Deja et al. [2023], and a Mixed-Accents branch for implementing our mixed accent speech synthesis model.

In Amazon-Reimplementation, the primary change to the Grad-TTS repository was incorporating accent embeddings. The Grad-TTS model optionally allowed the user to specify a speaker and incorporated speaker embeddings. So to incorporate the accent embeddings, we largely mirrored the existing code that used speaker embeddings. See for example in the Amazon-Reimplementation branch Grad-TTS/model/tts.py lines 48 and 79 where an accent embedding was created. There are small changes throughout the code to accommodate this but there are too many to list. We also included code for data preprocessing in Amazon-TTS /prepare_commonaccent.py. To find the accent label in training examples, we ran the CommonAccent accent classifier from Zuluaga-Gomez et al. [2023] from network on each audio sample and took the argmax.

In the Mixed-Accents branch, we modified the code from Amazon-Reimplementation. We replaced the accent id input with an accent spread input. And to get the accent embedding, we ran the accent spread through a feed forward neural network. You can see this on lines 20-35, 48, 96-103, and 154-160 of Grad-TTS/model/tts.py on the Mixed-Accents branch. We also preprocessed training data by downloading the audio, computing the accent spreads, formatting the examples into a readable text file in Grad-TTS/transform_accent.py in the Mixed-Accents branch.

We also wrote various pieces of testing code to calculate accent fidelity, produce the samples for the survey, and calculate text fidelity. See Testing.ipynb, CreateAmazonSamples.ipynb, CreateGradTTSSamples.ipynb, and CreateMixedAccentSamples.ipynb.

7 Timeline

The table below outlines the time spent by each team member on various components of the project. Hours are approximate.

Task	Shawn	Ben	Richard
Reading Papers / Dataset Research	10	10	10
Reading Code Documentation	4	4	4
Understanding GradTTS Baseline	2	2	2
Replicating GradTTS	9	4	1
Reimplementing Amazon TTS System	0	10	13
Accent Embedding Model Dev.	0	6	6
Modifying Conditioning Pipeline	0	6	6
Writing Scripts for Experiments and Running them	4	20	20
Writing Executive Summary	6	2	0

Table 2: Estimated hours spent per task by each team member

8 Research Log

The development of our accent-conditioned text-to-speech synthesis framework involved navigating multiple unforeseen challenges and adapting our strategy dynamically in response.

One of our earliest hurdles was replicating the baseline Grad-TTS model. Although the original Grad-TTS paper reported training for 10,000 epochs, significant computational resource demands and time constraints caused us to train for fewer epochs. We trained the GradTTS model for only 875 epochs as it quickly achieved high quality audio. We trained our Amazon reimplementation for 1200 epochs and our own mixed accent model for 1500 epochs. Each epoch took several minutes to complete, resulting in a total training time of nearly three days on one of our available GPUs (NVIDIA RTX 4080). We also trained on T4 and A100 GPUs on Google Colab where epochs took 1 to 2 minutes. Consequently, we could not achieve the full performance originally demonstrated by Grad-TTS, which somewhat limited the baseline fidelity and the robustness of subsequent comparisons.

Another particularly challenging aspect of the project involved creating accurate accent labels using an external accent classification model. Initial attempts to leverage existing classifiers encountered obstacles due to varying levels of accuracy, inconsistent performance, and compatibility issues. After some basic trial and error and experimentation, we eventually settled on the CommonAccent classifier from Zuluaga-Gomez et al. [2023], which reliably provided the accent distributions required for conditioning our generative model. Integrating this classifier effectively and ensuring the quality of the labels also required considerable effort. Switching from the CommonAccent corpus to Common Voice 12.0 substantially increased preprocessing time. The raw dataset is an order of magnitude larger and many clips lack explicit accent accent labels and none contain accent spreads, so we had to pipeline batch inference with the CommonAccent classifier and store both the *argmax* label and full probability vector. End-to-end preprocessing took roughly 11 GPU-hours and produced 74 GB of intermediate feature files.

One of the difficult coding challenges was the reimplementation of Amazon's diffusion-based accent synthesis model. Since Amazon's original paper did not include publicly available source code or explicit architectural details, Ben and Richard had to implement this model entirely from scratch based solely on textual descriptions from the paper. This lack of direct reference led to multiple iterations and many debugging sessions which increased our originally planned implementation timeline.

Our inference pipeline stability also presented notable complications. Running the inference pipeline to generate synthesized speech samples gave us many issues, including unexpected software dependencies, pipeline incompatibilities, and challenges exporting final synthesized audio outputs into usable audio formats (e.g., MP3). These difficulties directly delayed the generation of our survey audio samples, pushing back the timeline for obtaining the data for our perceptual naturalness results from human participants.

Finally, we encountered high volatility in the quality of synthesized speech outputs. Outputs varied notably across different runs, even under consistent input conditions. This instability complicated our evaluation process, as it made it challenging to consistently benchmark our mixed-accent model against our baseline implementations. Additionally, the variation and lack of distinct accent generation made our analysis and post-processing very difficult than initially anticipated.

Despite these setbacks, our iterative approach, consisting of frequent team meetings, pair programming, targeted debugging, and repeated experimental runs allowed us to overcome or mitigate many of these challenges. The obstacles encountered contributed to our understanding of the complexities inherent in developing generative audio models, particularly concerning conditioning mechanisms, accent embedding, and computational constraints in realistic research environments.

9 Conclusion

In this work, we introduced a new approach for generating text to speech audio that blends multiple accents naturally. Our method built upon the GradTTS framework by allowing users to specify desired accents through intuitive probability distributions. We trained our system using the Common Voice 12.0 dataset, extracting accent labels automatically through an external classifier.

Participants successfully identified accent blends with an accuracy of 64.4% and rated the quality of our synthesized speech at an average Mean Opinion Score (MOS) of 1.96. These findings suggest that our method conveys accent blends in generated speech.

However, we encountered challenges such as output quality variability and computational constraints, highlighting areas for further improvement. Future work could focus on stabilizing audio quality, enhancing inference efficiency, and exploring additional accent combinations. These advancements would improve the usability and realism of our approach, making text to speech systems more accessible and representative of global linguistic diversity.

References

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215, 2020.

Google Cloud. Cloud speech-to-text.

Katarzyna Deja, Eliya Nachmani, and Joseph Keshet. Diffusion-based accent modelling in speech synthesis, 2023.

Keith Ito and Linda Johnson. The lj speech dataset, 2017. URL `https://keithito.com/LJ-Speech-Dataset/`.

Jianzong Liu, Wenjie Wang, Shiyin Kang, Mengchun Zhang, and Jing Xiao. Controllable accented text-to-speech synthesis with fine- and coarse-grained intensity rendering, 2022.

Jan Melechovsky, Ambuj Mehrish, Berrak Sisman, and Dorien Herremans. Accent conversion in text-to-speech using multi-level vae and adversarial training. *arXiv preprint arXiv:2406.01018*, 2024a.

Jan Melechovsky, Ambuj Mehrish, Berrak Sisman, and Dorien Herremans. Dart: Disentanglement of accent and speaker representation in multispeaker text-to-speech. *arXiv preprint arXiv:2410.13342*, 2024b.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, Mikhail Kudinov, and Jun Wei. Grad-tts: A diffusion probabilistic model for text-to-speech, 2021.

Zhan Zhang and Jiaxing Chen. Accent recognition with hybrid phonetic features. *Sensors*, 21(18):6258, 2021. doi: 10.3390/s21186258.

Jinzuomu Zhong, Korin Richmond, Zhiba Su, and Siqi Sun. Accentbox: Towards high-fidelity zero-shot accent generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Xuehao Zhou, Mingyang Zhang, Yi Zhou, Zhizheng Wu, and Haizhou Li. Multi-scale accent modeling and disentangling for multi-speaker multi-accent text-to-speech synthesis, 2025. URL `https://arxiv.org/abs/2406.10844`.

Juan Zuluaga-Gomez, Sara Ahmed, Danielius Visockas, and Cem Subakan. Commonaccent: exploring large acoustic pretrained models for accent classification based on common voice. *arXiv preprint arXiv:2305.18283*, 2023.

A Survey Contents

For each audio sample in the audio quality section of the survey, users were asked to rate the samples on a scale from 1 ("Bad Quality") to 5 ("Excellent Quality") with half point increments. The five text samples were drawn uniformly at random from the LJ speech dataset Ito and Johnson [2017].

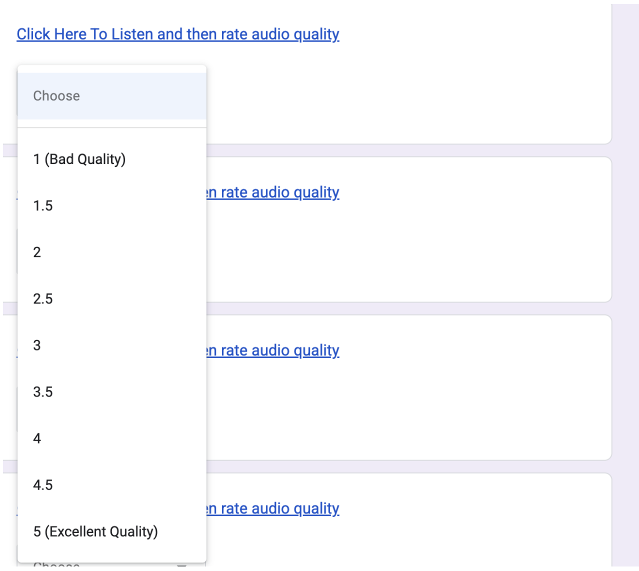

Figure 7: Google Form audio quality question example

In the accent classification sections, we used Google Forms' grid answer feature which allows us to limit responses so that users cannot classify the multiple clips from the same text prompt as the same accent.

For the accent classification questions, we used three text samples from the Universal Declaration of Human Rights:

- Everyone has the right to life, liberty and security of person.
- All human beings are born free and equal in dignity and rights.
- Everyone has the right to recognition everywhere as a person before the law.

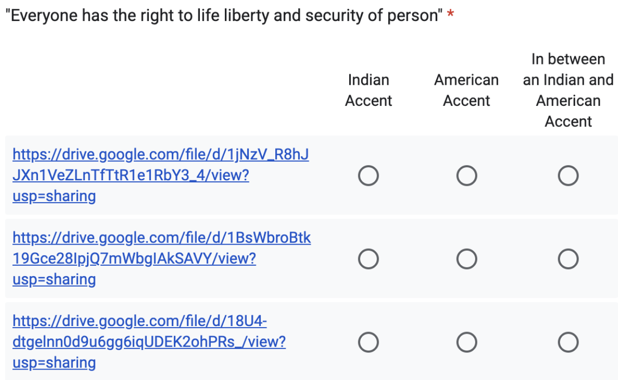

Figure 8: Mixed accent classification question example

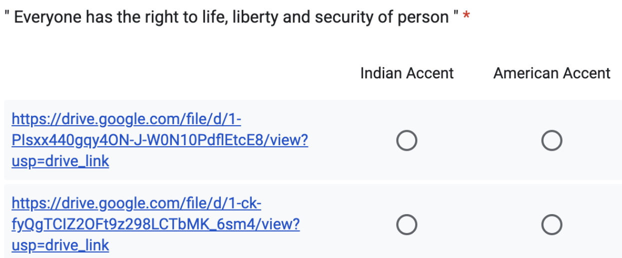

Figure 9: Amazon reimplementation accent classification question example