Voice Cloning Resume Website

Dade Wood[1], *Ken Taylor*[1], *Huan Ma*[1]

[1]Rochester Institute of Technology (RIT)

daw1882@rit.edu, kat6106@rit.edu, hm4329@rit.edu

Abstract

Voice cloning is a technique to take a few samples of an individual's voice and replicate it using synthetic speech. This has a wide range of use cases from personal assistants to helping the vocally handicapped. In recent years, the job market in the field of software development has become increasingly competitive, with personal websites and projects becoming more necessary to stand out. For this project, we propose using voice cloning techniques to create a simple resume chatbot website that will answer questions about yourself with your own synthesized voice. This proposed method gives us two research questions to answer. RQ1, how accurately will different speech synthesis models clone our voices, and will they perform this task sufficiently well with limited training data? And RQ2, does connecting a chatbot and a speech synthesis model create a realistic conversation-like experience for site visitors? Our minimum viable product is replicating a voice cloning model to clone a voice from a short set of recordings. Our extended goal is to combine part one with a conversation language model trained on a resume that answers questions with a small set of answers using the cloned voice.

1. Introduction

The task of speech synthesis involves creating audio files from text, by putting formants in the right places to make each phone, adjusting formants for coarticulation, and applying accents. The input is text, and the output is spoken audio. Text-to-speech (TTS) has been augmented by machine learning, but the core algorithm still relies on language analysis. We focus on a specific application of speech synthesis: voice cloning (VC). VC involves fine tuning a TTS on a specific individual's voice, with the goal of passing the generated audio off as authentic speech of the cloned individual. This is a useful tool in the 21st century, from video skits where political figures act out of character, to movies where a lead actor died during production, to recovering someone's voice after injuries to the vocal track. VC relies on finding the features that are unique to each speaker, so this technology is also related to other speech processing tasks such as speaker identification or fingerprinting. Challenges in voice cloning include feature selection; determining what characteristics are unique to the speaker, so the model doesn't waste resources on factors that are common to all speech.

In research sometimes there is value to be had in assembling components in novel ways, instead of always focusing on creating new technologies. Well-funded teams of tens of researchers have the resources to write huge codebases and train models on powerful hardware. Smaller research teams then have the opportunity to pick the tools that work the best for niche tasks of personal interest. We have experimented with natural language processing tools and decided to connect a few of them together to create an interactive resume to stand out to recruiters. Potential employers can type questions like: "Tell me about one time you overcame a challenge, and how you resolved it." and hear responses generated from the potential hire's own work experience, spoken in their own voice. The experience resembles a mock interview. It is mostly for novelty, but like any portfolio project, its function is to demonstrate competency in the involved programming concepts.

2. Related Work

Voice cloning refers to the process of generating a specified human voice sample by using synthetic language techniques. In recent years, with the rise of machine learning and deep learning methods, sound cloning technology has made great progress. This literature review will provide an overview of the techniques involved in voice cloning.

2.1. Speech synthesis

Speech synthesis is the process of generating synthetic speech from text using various statistical, machine learning, or deep learning techniques. In Oord et al., Wavenet is proposed as a deep neural network that generates original audio waveforms [1]. This model is able to sound more natural than splicing systems. Arik et al. designed an end-to-end neural voice based entirely on deep neural networks [2]. The synthetic model, compared with a traditional TTS model, does not require feature engineering and has a faster training speed. Ping et al. proposed a fully-convolutional attention-based neural TTS system [3]. Compared with the previous deep neural network [2], this model retains the advantages of fast computing speed while it also expands the query scale on the GPU. Taigman et al. demonstrated a new TTS method that can handle unconstrained voice samples and does not need to align phonemes and other language features [4]. Additionally, this method can also convert the speaker's identity with a small number of samples.

2.2. Voice conversion

Voice conversion is the process of changing the voice audio of a source speaker to match the voice of some target speaker. Desai et al. proposed a method of using ANN to capture speaker features, which is suitable for converting between single or multiple languages [5]. Chen et al. proposed a sound conversion system based on deep neural networks [6]. This model constructed a nonlinear mapping relationship between speaker spectra. Experiments have proved that compared with traditional GMM, it has better similarity and naturalness. A probabilistic interpretation for ANN-based VC is proposed by Hwang et al., achieving the same performance as the state-of-the-art GMM-based VC with the MLGV algorithm [7].

2.3. Voice Cloning

Voice cloning can be seen as the intersection between speech synthesis and voice conversion as it is the process of generating synthetic speech that replicates the voice of a target speaker. This task is also often done with the additional goal of cloning the speaker's voice using only small amounts of data. In Arik et al., the authors showcased two voice cloning methods and find that speaker adaption performs slightly better in naturalness and similarity but speaker encoding performs better in cloning time and required memory [8]. Luong and Yamagishi proposed NAUTILUS, a speech synthesis system capable of modifying the cloning strategy depending on additional data to provide more focus on TTS or voice conversion [9]. Xie et al. provided an outline of the Multi-speaker Multi-style Voice Cloning Challenge (M2VoC) of 2021 which includes a summary of evaluation methods for voice cloning and submitted systems to the challenge [10]

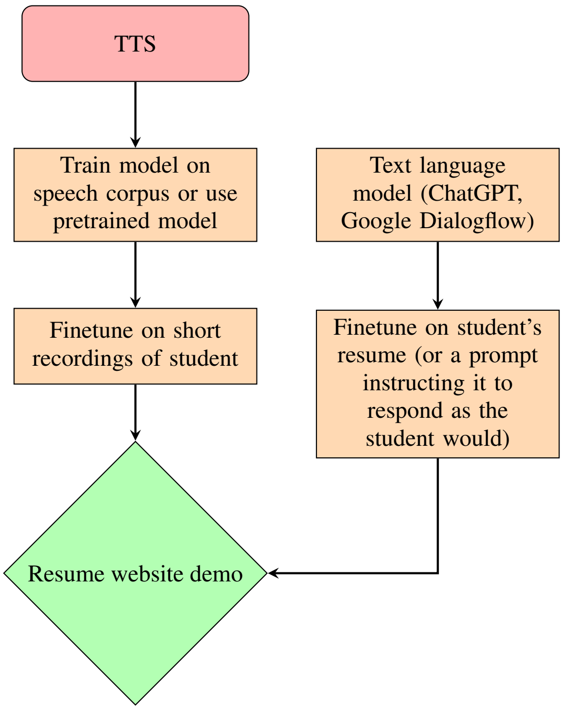

Figure 1: *Flowchart of our methodology process.*

3. Methodology

3.1. Overview

The text-to-speech model and its fine-tuning are the core of our project. We have selected a few example papers that have pre-trained TTS models. To select the model to use, we are recreating the methods described in the papers we selected, downloading the pre-trained models and/or source code they provide, and running them for evaluation. Some papers don't provide source code or executables, so although they may show promising performance, they are not able to be used for this project. Below are the three models we chose to implement and test for this project:

- **Tortoise TTS** (Tortoise) [11] is an open source text-to-speech, voice cloning system created in Pytorch.
- **Real Time Voice Cloning** (RTVC) [12] is an open source Pytorch implementation of the SV2TTS system by Jia et al [13].
- **Neural Voice Cloning with Few Samples** (NVCFS) [14] is an open source Pytorch implementation of the voice cloning paper with the same name by Arik et al. [2].

With the 3 models outlined above, we performed evaluations and chose Tortoise as our main model for our website demo. Tortoise is then connected to the text output of a conversational model through the use of a REST API. We choose ChatGPT for our conversational model which is prompted by the target speaker's work experience and resume. Alternatively, other models such as Google Dialogflow or custom models without deep learning could be implemented as conversational, our focus is primarily on the VC system.

Once we have both a voice cloning model and a conversational model, we then combine the two through the use of APIs to create a resume website experience that will answer questions about an individual's resume in their voice using as few as one reference sample of audio.

3.2. Data Creation

For our voice cloning models to work, we need audio samples from a target speaker. We use the methodology and transcribed sentences as described in Arik et al. [8] to generate our target speaker samples. Recordings were done in a quiet room on a laptop microphone all in one sitting. After recording, we have 30 audio samples of three different target speakers for inference and evaluation. Target speakers are chosen to be the authors of this paper, Dade, Huan, and Ken.

3.3. Implementation

3.3.1. Voice Cloning Models

RTVC was cloned and run locally according to the instructions in the readme. As input, it takes in only a single audio clip of the target speaker to perform voice cloning. The selected sample for input is the audio clip of the longest sentence from our transcribed sentences.

NVCFS provides no pretrained models for use and requires the training of a new model. Because this training is very time consuming, even on a GPU, we decided to do our subjective evaluations on the audio samples provided by the authors instead of our own. The original audio samples of the target speakers could not be found so our evaluation of it was further limited.

Tortoise was first implemented and tested on Google Colab using a GPU runtime by cloning the repository and running it according to instructions in the readme. For our official demo, we implement this model in a Hugging Face Space, a useful online tool for creating machine learning demos, and created a REST API to call and perform inference jobs on it. The Space runs on the Nvidia T4 small GPU setting. The model is able to take in a varying number of samples of the target speaker as input, we choose the audio clips of the 3 longest sentences from our data as Tortoise recommends a minimum of 3 samples for good results. We run all inferences using the *ultra_fast* setting of Tortoise which provides the worst quality results but has the fastest run-time.

Table 1: *5-point MOS for TTS quality metrics. Bold face scores show the model with the best score in each dimension. As shown, the Tortoise model has the highest MOS for all categories.*

Model	Naturalness	Intelligibility	Expressiveness
Tortoise	**3.33**	**3.44**	**3.00**
RTVC	2.33	3.33	**3.00**
NVCFS	1.56	2.22	2.67

3.3.2. ChatGPT

We selected ChatGPT for the chatbot component because of familiarity with the tool and its capabilities, and because the API was already up and running for it, letting us focus more effort on voice cloning. We created a ChatGPT prompt that would give the ai enough information about job history to impersonate us: *resume.txt + I want you to answer questions as if you were me, using the information in my resume*. Then, we connected to the API to send this long prompt and receive an answer. For the demo, we limited responses to two sentences because of how long the voice cloning was taking. One consideration was whether including the resume in the prompt would take up too much of the max token count, but after initial testing, we found that the 4096 token cap was plenty, as we only used 700 for a resume, prompt, and response. ChatGPT token counts can be estimated with word counts, and once the request is made, the actual total number is returned with the response.

3.3.3. Website Demo

We used the python *simplehttpserver* package to collect user questions from a web form, then display the ChatGPT response in text form while playing the cloned audio. Achieving the ChatGPT functionality was simple: concatenate the user question to the resume and prompt, and call the API. The string returned was then split into sentences, and sent as separate calls to our Hugging Face API. If more than about one sentence was sent at once, the model could use too much memory or produce poor results. We aren't sure why, but this was a problem with at least one other TTS we tried besides Tortoise. The server collects each mp3 file, writes them into one long one, and sends it to the browser. For two sentences, with Hugging face on gpu mode, this takes about a minute. The python code for this server is made available in the project github.

4. Evaluation

4.1. Subjective VC model evaluation

We conducted listening tests to evaluate the performance of three voice cloning models on the following five dimensions:

1. **Intelligibility**: how understandable the cloned voice is.
2. **Naturalness**: how natural sounding the cloned voice is.
3. **Expressiveness**: how expressive in terms of pitch, tone, etc. the cloned voice is.
4. **Speaker Similarity**: How similar the cloned voice sounds to the target voice.
5. **Style Similarity**: How similar the style of the speaker sounds to the original voice (e.g. pauses, cadence, tone, etc.).

We perform a subjective assessment for both quality of the TTS as well as voice cloning performance. All metrics are scored by the authors of this paper as participants. TTS quality evaluation is performed by having the participants listen to a total of 9 anonymous and randomized audio samples, 3 from each model being evaluated, and rate each sample on Intelligibility, Naturalness, and Expressiveness. After quality evaluation, we continued to voice cloning evaluation with only the Tortoise and RTVC models as they had the best quality scores and did not require long training times like NVCFS. Voice cloning is evaluated on speaker and style similarity for each of the three target speakers as well as overall performance. Participants evaluated by listening to original samples from the target speaker and compare to anonymous and randomized samples from both models speaking the same sentence as in the original sample.

Table 2: *5-point MOS for voice cloning metrics. Bold face scores show the model with the best score in each dimension. Tortoise performed noticeably better for Ken and Dade, but the results were mixed for Huan.*

Target Speaker	Speaker Similarity	Style Similarity
Dade		
Tortoise	**3.53**	**3.60**
RTVC	2.27	2.47
Huan		
Tortoise	1.8	**1.87**
RTVC	**2.47**	**1.87**
Ken		
Tortoise	**3.47**	**3.27**
RTVC	2.07	2.80
Overall		
Tortoise	**2.93**	**2.91**
RTVC	2.27	2.38

The score range for all metrics is 1-5, 1 point for bad, 2 points for poor, 3 points for fair, 4 points for good, and 5 points for excellent. All participants' scores are taken as an arithmetic mean to get each model's Mean Opinion Score (MOS) in each dimension.

The results in Table 1 show that the Tortoise model had the best scores across all dimensions for quality while NVCFS had the worst. RTVC had the same MOS for Expressiveness as Tortoise and was only 0.11 worse in Intelligibility but was outperformed in Naturalness by a full point. RTVC and NVCFS both suffer from high degrees of noise in their outputs that result in voice audio that sounds very robotic or fake but Tortoise is able to produce output with much less noise, which explains these results.

Table 2 shows that Tortoise significantly outperforms RTVC in MOS for the cloning metrics for both of the native English speakers, Dade and Ken, but for non native English speaker, Huan, the results were much more mixed and RTVC performed better for speaker similarity. This is likely due to RTVC's method of simply applying a speaker embedding to the model output to make it sound closer to the target speaker while Tortoise is a much more complex model trained on hundreds of hours of audio that could be affecting the sound of the final output if the data trained on is biased towards native speakers. Ultimately, Tortoise was chosen to proceed with the demo, because it performed the best overall.

4.2. Conversation Model

ChatGPT does a good job of impersonation most of the time. It creates sentences that are relevant to both the question and the resume. For example, *My last job was as a Machine Learning Engineer Intern at COMPANY* in Table 2, Response 1 came directly from the resume and therefore is truthful. However, it can also generate statements that are not true, although responses are limited to ideas relevant to the prompt. For example, the language model hallucinates extra information, such as *transfer learning and pruning* (Table 2, Response 2), neither of which are present in the resume.

Table 3: *ChatGPT responses to employer questions, impersonating Dade using his resume. Some hallucinations are present in Response 2 while Response 3 answers from the perspective of ChatGPT. Response 1 had the best impersonation results. Workplace names have been replaced with COMPANY for privacy.[15]*

Question	Response
(1) What was your last job?	My last job was as a Machine Learning Engineer Intern at COMPANY (formerly known as COMPANY), where I developed object detection solutions in Python for images captured from drones and worked on analyzing wind turbine damages.
(2) What was one time you encountered a problem, and how did you overcome it?	During my internship at COMPANY, I encountered challenges with optimizing the accuracy and speed of a machine learning model for detecting wind turbine damages. I overcame this by implementing various techniques such as transfer learning and pruning, while also collaborating with the cloud team to improve the model's performance.
(3) What was one valuable skill you learned from your time at COMPANY?	As Dade Wood, I learned how to collaborate in a small team to design, develop, and implement a novel end-to-end model in Python that combines the benefits of a convolutional neural network and the k-nearest neighbors algorithm.

Our ChatGPT evaluation is very subjective. On the one hand, hallucinations introduce information that wasn't in the prompt and is technically not true. On the other hand, this is the behavior we wanted to see when we chose to use ChatGPT. The task of creating an anecdote from a single page bullet-point list of experiences is essentially a hallucination. In this aspect, the responses we produced are close to our goal. In our experiments, we found that tuning the *temperature* hyperparameter closer to 1 gave more interesting responses. Turning temperature down didn't remove the false details, but made the embellishments more boring and vague. One aspect the results have much more shortcomings in is "role". In our testing, we found that the responses would sometimes take the role of "ChatGPT" as speaker, instead of the prompt of "impersonating Dade". The language model claims to have been Dade in the past tense, which is something the real Dade would never say. Our prompt gave inconsistent impersonation results, but could be improved by making the original request more strict, for example, by prompting with "respond only exactly how Dade would speak to an employer in an interview."

5. Discussion

5.1. Model Efficiency

From Tortoise's github repository, the tagline of this model is *it's called Tortoise because it's insanely slow* [11]. This means that the run-time of this model for inference can be in the order of hours for CPU and the order of minutes on GPU, depending on settings. We use the least accurate, most efficient setting of *ultra_fast* for all of our audio generations, and even on a GPU it can take close to a minute to complete inference, depending on the length of the prompt. These run-times and resource requirements are important to consider for a demo such as this one as it affects the accessibility and realism of the system.

5.2. Ethical Considerations

Due to the highly controversial nature of cloning a person's voice, it would be prudent to discuss the ethical concerns that should be considered for this project.

5.2.1. Voice Cloning

Voice cloning is a technology such that, if used in a malicious manner, can cause great harm. For example, a cloned voice could be used to impersonate someone and make it sound as if they said something discriminatory, hateful, or harassing. For this reason, it is important that the development of voice classification and identification be prioritized in order to detect generated audio and flag it. It is also highly important that the consent of an individual is acquired before cloning and using their voice for this technology.

5.2.2. ChatGPT

ChatGPT is trained on millions upon millions of different texts from all over the internet. As a result, this means that the inferences it makes for its chats can sometimes be influenced by information completely separate from the prompt provided. This means for a project such as this, where we have made a website for showcasing your resume and work experience, ChatGPT can sometimes hallucinate or make up information that is not true. Thus, to prevent lying to potential employers, a disclaimer should be given that responses generated could contain false or exaggerated details as created by the conversation model. Alternatively, a more simplistic implementation such as Google DialogFlow could be used in which a number of prompts results in previously written, truthful, responses.

5.2.3. Resources

Like with many large deep learning models, the Tortoise model we selected for this project is highly resource intensive. If used on a CPU it can take up to an hour to generate a response and can even take a few minutes on a GPU, depending on settings, and it can use up to 16GB of memory for the higher quality voices. This raises concerns about the impact models such as this one can have on the environment as the high computing usage results in lots of energy being used. To reduce this negative impact, more efficient models should be chosen or the Tortoise model should be further improved to use it's resources more efficiently.

6. Conclusion & Future Work

With the development of speech technology, more and more speech processing applications are being used in daily life from personal assistants to customer service dialogue systems. In this article, we use the subjective evaluation method MOS to select the voice cloning model Tortoise, combine it with the conversation model ChatGPT, and finally apply this composite model to a website for the purposes of answering questions about an individual's resume in their voice. This process can easily be replicated by others using the Hugging Face API we have created for the Tortoise model and ChatGPT's own API. Using this method, users can create a website that showcases their work and gives recruiters a unique experience for interacting with an individual's resume.

To answer our research questions, we were able to accurately clone our voices withe the pretrained models provided in TTS papers, and we were not able to create a fully conversation-like experience because of the latency between turns.

This method currently has two main limitations in the efficiency of Tortoise as well as the size of our prompt to ChatGPT. Future work includes implementing and evaluating more voice cloning methods, expanding our evaluation to include objective measurements such as Speaker Classification and Speaker Identification, and modifying our prompt to ChatGPT to be more concise and restrictive.

7. References

[1] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep voice: Real-time neural text-to-speech," in *International conference on machine learning*. PMLR, 2017, pp. 195–204.

[3] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[4] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv preprint arXiv:1707.06588*, 2017.

[5] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3893–3896.

[6] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[7] H.-T. Hwang, Y. Tsao, H.-M. Wang, Y.-R. Wang, and S.-H. Chen, "A probabilistic interpretation for artificial neural network-based voice conversion," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 552–558.

[8] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," *Advances in neural information processing systems*, vol. 31, 2018.

[9] H.-T. Luong and J. Yamagishi, "Nautilus: A versatile voice cloning system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2967–2981, 2020.

[10] Q. Xie, X. Tian, G. Liu, K. Song, L. Xie, Z. Wu, H. Li, S. Shi, H. Li, F. Hong, H. Bu, and X. Xu, "The multi-speaker multi-style voice cloning challenge 2021," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8613–8617.

[11] J. Betker, "TorToiSe text-to-speech," Apr. 2022. [Online]. Available: https://github.com/neonbjb/tortoise-tts

[12] C. Jemine, "Real time voice cloning," 2019. [Online]. Available: https://github.com/CorentinJ/Real-Time-Voice-Cloning

[13] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," 2019.

[14] S. Chitlangia, M. Rastogi, and R. Ganguly, "An open source implementation of neural voice cloning with few samples," Available at https://github.com/SforAiDl/Neural-Voice-Cloning-With-Few-Samples/ or https://github.com/Sharad24/Neural-Voice-Cloning-With-Few-Samples/, 2021.

[15] OpenAI, "Chatgpt [large language model]," Available at https://chat.openai.com/chat, 2023.

[16] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[17] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[18] P. Neekhara, S. Hussain, S. Dubnov, F. Koushanfar, and J. McAuley, "Expressive neural voice cloning," 2021.