VoiceChat - Bringing chats to life using Deep Learning

Laxman Vikas Kommireddi
CSE - AIML
MLR Institute of Technology
Hyderabad, India
vickyvikas463@gmail.com

Yogeshwar Rayudu
CSE - AIML
MLR Institute of Technology
Hyderabad, India
yogirayudu@gmail.com

Pavan Kalyan Aouti
CSE - AIML
MLR Institute of Technology
Hyderabad, India
pavankalyanaouti@gmail.com

Mohammed Ikramuddin
CSE - AIML
MLR Institute of Technology
Hyderabad, India
ikrammohd109@gmail.com

Mr. Sai Prasad Kashi
HOD of CSE(AIML)
MLR Institute of Technology
Hyderabad, India

Mr. V S. Pavan Kumar
Assistant Professor
MLR Institute of Technology
Hyderabad, India

ABSTRACT

This project describes a revolutionary chat application that leverages efficient voice cloning technology to revolutionise digital communication. With the help of this invention text messages can be now personalised and sent as audio messages with the sender's voice. The proposed system captures the authenticity of the sender's speech by smoothly integrating AI algorithms to mimic vocal patterns and nuances. This study develops a trustworthy and adaptable voice replication system by implementing developments in neural networks and speech processing. Using a combination of deep neural networks and chat application development, the project aims to analyse and capture detailed human vocal patterns. The algorithm extracts the subject's vocal characteristics, intonations and speech patterns using as few human voice samples as possible. This method enhances communication quality in general and emotional expression. The application aims to combine application development and deep learning with advanced options, pushing the boundaries of technical innovation. This synopsis focuses on the creation of bridging the text-voice divide, enhancing the accessibility, personalisation, and engagement of digital communication.

0.1 CCS Concepts

Chat application, Deep Learning, TorToiSe

0.2 Keywords

Application Development, Deep learning, Text-to-speech, Voice Cloning

1 INTRODUCTION

An application is being developed whose core technology relies on [1] voice cloning techniques so that when you write a text message to someone, it plays back as an audio file. For a text message, when the user chooses the recipient, he can then activate his kind of voice. Using existing [2] voice samples, the algorithm captures the sender's unique voice patterns. Through the end user's input, an artificial voice that closely matches the sender's original vocal characteristics is created by sophisticated AI algorithms. The result is an audio message that both preserves the expressiveness and intimate feel of the original speaker. In voice cloning, [21] deep neural networks are employed Generating human-like copies of their voices by voice cloning is an interesting technique that turns human text into speech. The ready availability of huge sound corpora and advances in deep learning algorithms have driven rapid development in this latest technique for generating sound. Thanks to improved deep learning algorithms, particularly those based on [6] convolutional neural networks (CNNs) and recurrent neural networks (RNNs)], it is now possible to analyze and generate sound in much more intricate ways. [3] Neural text-to-speech (TTS) models, however, can now produce such speech which is very much like human speech.

TorToiSe is one such algorithm which shows promising results in voice cloning. However, as the model progresses, it gradually gets to replicate the person's regional accent, inflections, grammar, and other signs of his identity. As an additional matter, the technology industry increasingly calls for personalization of the user experience. Since the use of messaging apps has grown so common, the idea of adding voice to text chat was born. Voice cloning could ultimately become a new and creative method for users to get messages across, bridging the gap between writing speech altogether. The program adopts a strong encryption mechanism to protect sensitive data such as voice recordings, so user data privacy can be guarded. Under the agreement of the user's prior consent will this voice be authenticated, showing that all laws and principles of conduct have been observed. When the cloned voice's authenticity is being tested, the program now constantly improves its models through user suggestions ratings, and content updates. To sum up: The imitation voice grows better and better towards what it is pretending to be over time because of this ongoing iterative improvement. That said, the combination of [9] speech synthesis, and [20] deep learning, the effect of combining these elements is a program that represents a genuinely novel communication medium by joining the user's distinctive voice with a synthesized speech that synchronizes perfectly with his writings. The proposed solution is for those people who desire to communicate more authentically. This chat application can be further developed in the future to include advanced features and functionalities like security, privacy and emotional expression. This ground-breaking technology promises a transfiguring communication and is expected to turn the future of digital voice conversations from faceless, standardized exchanges marked by machine calligraphy into a world where small user interfaces, AI algorithms, and [17] voice cloning merge seamlessly.

2 BRIEF INTRODUCTION OF TTS

Text-to-speech (TTS) Figure 1 is a technology that converts written text into spoken words. It enables computers, devices, and [19] applications to produce natural-sounding speech output. TTS systems analyse the input text, apply linguistic rules, and use synthesized

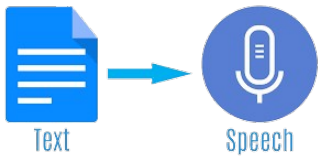

Figure 1: Text-to-speech

voices to [7] generate speech. These systems have numerous applications, including accessibility features for individuals with visual impairments, language learning tools, navigation systems, and automated customer service. Advances in TTS technology have led to more natural-sounding voices and improved intelligibility, making them increasingly prevalent in everyday devices and services.

3 EVALUATION OF TTS

Evaluating Text-to-Speech (TTS) systems involves assessing various aspects of their performance, including: Naturalness of Speech: One of the most crucial aspects is how natural the synthesized speech sounds. Listeners should perceive the speech as human-like, with appropriate intonation, rhythm, and pronunciation. Intelligibility: Intelligibility measures how easily listeners can understand the synthesized speech. Factors such as clarity, articulation, and pronunciation accuracy contribute to intelligibility. Prosody and Emotion: Effective TTS systems can convey the appropriate emotional tone and emphasis in the synthesized speech. Prosody refers to features like pitch, stress, and rhythm, which play a significant role in conveying meaning and emotion. Linguistic Accuracy: TTS systems should accurately pronounce words and handle linguistic nuances such as word stress, intonation patterns, and proper names. Errors in pronunciation can significantly affect comprehension. Adaptability and Customization: Evaluation may consider how well TTS systems adapt to different contexts, accents, and [26] languages. Customization options such as voice selection and speech rate can enhance user experience. Latency and Efficiency: Latency measures the time delay between input text and output speech. Efficient TTS systems should produce speech output promptly without significant delays or computational overhead. Robustness and Error Handling: Evaluators may assess how TTS systems handle input errors, ambiguous text, or uncommon words. Robust systems should gracefully handle such situations without producing unintelligible or unnatural speech. User Satisfaction: Ultimately, user feedback plays a crucial role in evaluating TTS systems. Surveys, user studies, and usability tests can gauge user satisfaction, preferences, and perceived quality of synthesized speech. Evaluation methodologies may include subjective assessments by human listeners, objective measurements using metrics like Word Error Rate (WER) or Mean Opinion Score (MOS), and real-world usage testing in relevant applications or scenarios. Continuous evaluation and feedback are essential for improving TTS systems and ensuring they meet the needs and expectations of users.

4 EXISTING PROBLEM

In the contemporary digital communication landscape, there exists a significant accessibility gap for individuals who are engaged in activities that demand hands-free interaction with messaging platforms. The conventional text-based nature of communication platforms poses a considerable barrier for these users, hindering their ability to receive and respond to messages promptly. Also, when messages don't feel personal or real in texts, it can make it hard to understand feelings, making the overall experience not as good. This happens a lot when you need to pay attention right away, like when driving or multitasking, where manually reading texts can be difficult and not safe. To address this challenge, a chat application should be developed that utilizes voice cloning technology and [11] deep learning to convert text messages into personalized audio messages in the sender's voice. The aim is to achieve accurate [13] voice cloning and provide lifelike and engaging audio results. The application aims to bridge the gap between text and voice, enhancing digital communication by adding a more personal and accessible dimension. To build a [4] chat application that brings chats to life, the following issues are considered:

- Lack of Accessibility:
  The current system lacks accessibility features for visually impaired users and elders who have age-related vision impairments or cognitive decline, making it challenging for them to read text messages. This limitation excludes a significant portion of users who rely on auditory assistance to consume information effectively.

- Limited Multitasking Capability:
  Users are unable to read messages while engaged in other activities, such as driving or exercising. This restriction forces users to stop what they are doing to read each message individually, potentially leading to distractions and safety concerns in certain situations.

- Inconvenience in noisy environments:
  Users find it difficult to send audio messages in environments like buses or markets which hinders their chatting experience.

- Inconvenience in reading lengthy messages:
  Users often feel reading lengthy messages is tiresome, whereas the proposed application can help read out the message while the user continues their work.
- Reduced User Engagement:
  Plain text communication may not be as engaging as direct communication.

5 IMPLEMENTATION

When a user receives a text message within the app and selects the chat, they can choose to utilize the text-to-speech feature.

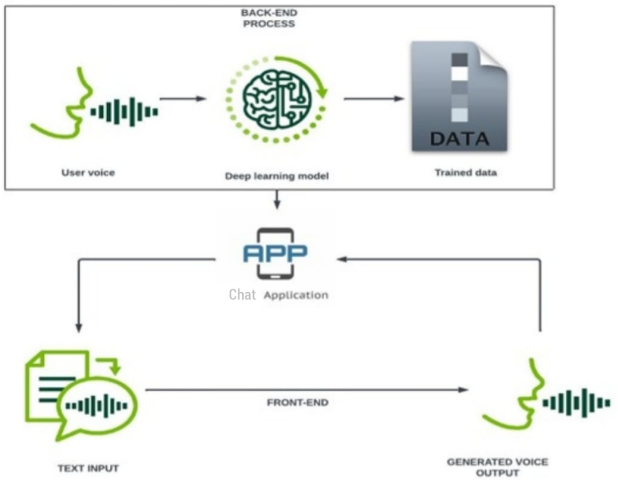

Figure 2: Implementation

(1) Recording User's Voice:
This step is the base for [16] voice cloning process. Users can register their voice to the application via recording or uploading an audio file.
(2) Analyzing voice patterns:
The process begins with the [23] app analysing existing voice recordings of the sender to capture their distinct vocal patterns, intonations, and nuances.
(3) Voice cloning:
These acoustic features are then fed into sophisticated AI algorithms, which generate a synthetic voice that closely emulates the sender's authentic voice.
(4) Integration of synthesized audio with text:
This synthesized voice is seamlessly applied to the text message, resulting in an audio message that retains the [5] emotional depth and personal touch of the original speaker.

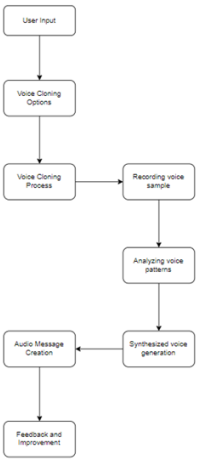

Figure 3: Workflow of VoiceChat

5.1 Pseudo code for TorToiSe

```
Define a parser to handle command-line
arguments

Parse command-line arguments to get options
such as text, voice selection, output options,
etc.

Load available voices and handle voice
selection

If text is not provided as command-line
arguments, read it from stdin

Split the text into manageable chunks if
needed

Set up settings for text-to-speech
synthesis including seed, preset, tuning
options, etc.

For each selected voice:
    Load voice samples and condition latent
    For each text chunk:
        Generate audio based on text and
        voice using the settings
        Save or play the generated audio
        depending on output options

    If the produce debug state option is
    enabled, save debugging information

Exit the program
```

The pseudo-code outlines the working of a text-to-speech (TTS) system implemented through a Python script with command-line interface. Here's a brief explanation of its operation:

- Argument Parsing: The script starts by parsing the command-line arguments using a predefined parser. These arguments specify details such as the input text, voice selection, output options, and other settings for [8] TTS synthesis.
- Voice Management: It loads available voices and handles voice selection based on the provided arguments. If [14] multiple voices are selected, it prepares to synthesize speech using each voice.
- Text Handling: If the input text is not provided through command-line arguments, the script reads it from the standard input (stdin). It also handles the splitting of the text into manageable chunks if needed.
- TTS Configuration: The script sets up the settings required for TTS synthesis, including parameters such as random seed, preset quality, and tuning options for the synthesis process.
- Synthesis Loop: For each selected voice and each text chunk, the script generates audio based on the text and voice using

the specified settings. It synthesizes speech for all combinations of voices and text chunks.
- Output Handling: Depending on the output options specified, the script either saves the synthesized audio to files, plays it back, or both. If multiple voices or text chunks are involved, it organizes the output accordingly.
- Exit: Once the synthesis process is complete, the script exits, concluding its operation.

5.2 Pseudo code for JavaScript

Algorithm: [12] Speech Synthesis and Chat Application

```
Algorithm: Speech Synthesis and Chat
Application

Initialize variables:
    txtInput, voiceList, btnSpeak: DOM
    elements for input text, voice selection,
    and speak button
    synth: SpeechSynthesis object
    voices: array to store available voices

Define function PopulateVoices():
    Retrieve available voices using
    synth.getVoices()
    Clear voiceList
    Iterate over available voices:
        Create option element for each voice
        in voiceList
        If it's the 45th voice, select it
        after a short delay
    Set selectedIndex of voiceList
    Hide voiceList dropdown

Define event listener for btnSpeak click:
    Create SpeechSynthesisUtterance object
    with input text value
    Find the selected voice from voiceList
    and assign it to toSpeak.voice
    Speak the utterance using synth.speak()

Define event listener for chatMessages click:
    Read the clicked message text
    using SpeechSynthesisUtterance and
    selected voice

Define function createChatMessageElement
(message):
    Create HTML message element using
    message data

Define window.onload event handler:
    Render existing messages in chatMessages
    using createChatMessageElement()

Define function updateMessageSender(name):
    Update messageSender variable
    Update chatHeader and chatInput
    placeholders based on messageSender
    Toggle active state for yogiSelectorBtn
    and vickySelectorBtn

Define event listeners for yogiSelectorBtn
and vickySelectorBtn clicks:
    Call updateMessageSender() with
    respective sender name

Define sendMessage function(e):
    Prevent default form submission behavior
    Generate timestamp
    Create message object with sender, text,
    and timestamp
    Save message to local storage
    Add message to DOM
    Clear input field
    Scroll chatMessages to bottom

Define event listeners for chatInputForm
submit and clearChatBtn click:
    Call sendMessage and clear local storage
    respectively
```

6 SOLUTION TO THE PROBLEM

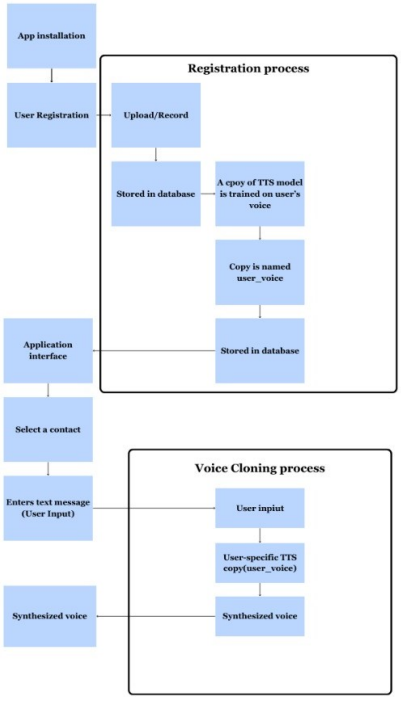

Figure 4: Architecture

As mentioned in Figure 3 The invention is a chat application that integrates cutting-edge voice cloning technology with [10] messaging functionality. It enables users to convert text-based chats into [22] audio messages delivered in the sender's voice. This innovation addresses the limitations of traditional text messaging by infusing conversations with a personalized and [18] emotionally rich dimension. It promotes accessibility for visually impaired individuals and aligns with the evolving landscape of digital communication. The project leverages advancements in AI, voice cloning, and chat [15] application development to create a revolutionary tool that transforms how people connect and converse in the digital age.

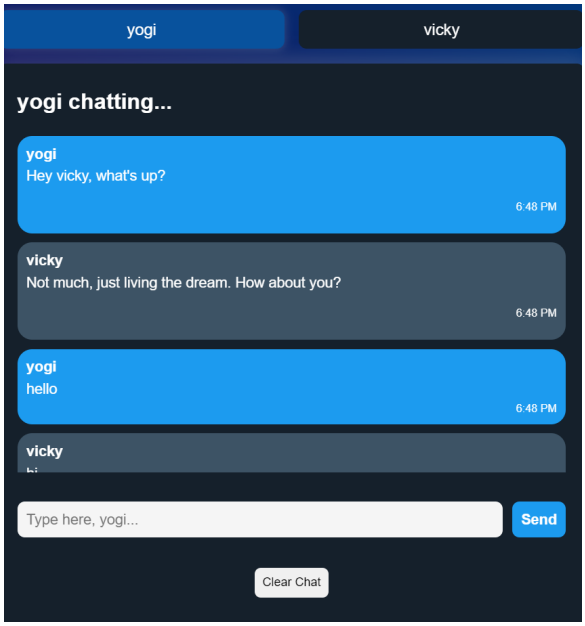

Figure 5: Live Implementation of application

Figure 5 presents the live implementation of the proposed solution, VoiceChat. In the above figure, we can see two users, yogi and vicky chatting with each other in the interface. The chats act as buttons and when clicked provide accurate text-to-speech synthesis. These chats are integrated with live time features to resemble traditional chat applications.

7 RESULT ANALYSIS

Figure 6 depicts the result analysis of the TorToiSe text-to-speech Algorithm. The analysis mainly focuses on

- alignment score
- loss decoder
- postnet parameters

8 CONCLUSION AND FUTURE SCOPE

Text-to-speech (TTS) is a technology that converts written text into spoken words. It enables computers, devices, and applications to produce natural-sounding speech output. TTS systems analyze the input text, apply linguistic rules, and use synthesized voices to generate speech.

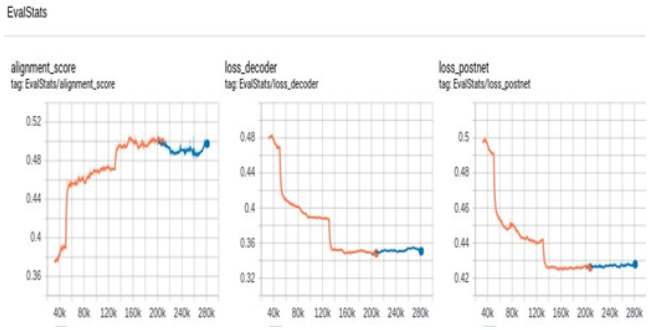

Figure 6: Performance metrics of Text to Speech Algorithm

These systems have numerous applications, including accessibility features for individuals with visual impairments, language learning tools, navigation systems, and automated customer service. Advances in TTS technology have led to more natural-sounding voices and improved intelligibility, making them increasingly prevalent in everyday devices and services.

This chat application can be developed further in the future which involves the following features:

- Advanced Deep Learning Algorithms: Better algorithms might evolve which can result in accurate [25] voice cloning, higher inference speed and lower response time.
- End-to-end security: [24] Security can be embedded into the chat application using encryption algorithms such as E2EE(end-to-end encryption).
- Added emotions: [27] Emotions can be added to the messages to make the messaging functionality emotionally rich.

REFERENCES

[1] Paarth Neekhara Shehzeen Hussain, Shlomo Dubnov, Farinaz Koushanfar, Julian McAuley, "Expressive Neural Voice Cloning" Feb 2, 2021.
[2] Sercan Ö. Arık , "Neural Voice Cloning with a Few Samples", 12 Oct 2018.
[3] Mike Chrzanowski, "Deep Voice: Real-time Neural Text-to-Speech", 2017.
[4] Akshata D Vhandale, "AN OVERVIEW OF REAL-TIME CHAT APPLICATION", 2022.
[5] Daria Diatlova, "EmoSpeech: Guiding FastSpeech2 Towards Emotional Text to Speech", 2023.
[6] Kainan Peng, Andrew Gibiansky, Sercan O. Arık Ajay Kannan, Sharan Narang,Wei Ping, "DEEP VOICE 3: SCALING TEXT-TO-SPEECH WITH CONVOLUTIONAL SEQUENCE LEARNING" , 22 February 2018.
[7] Aaron van den Oord, Sander Dieleman, Heiga Zen , Karen Simonyan Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu,"WAVENET: A GENERATIVE MODEL FOR RAW AUDIO" 19 september 2016.
[8] Nal Kalchbrenner, Erich Elsen,Karen Simonyan,Seb Noury, Norman Casagrande, Edward Lockhart , Florian Stimberg , Aaron van den Oord, Sander Dieleman, Koray Kavukcuoglu," Efficient Neural Audio Synthesis"25 June 2018.

[9] Yuxuan Wang , RJ Skerry-Ryan , Daisy Stanton, Yonghui Wu, Ron J. Weiss , Navdeep Jaitly, Zongheng Yang, Ying Xian , Zhifeng Chen, Samy Bengio , Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous,"TACOTRON: TOWARDS END-TO-END SPEECH SYNTHESIS " 6 april 2017.
[10] Raman Singh, Ark NandanSingh Chauhan, Hitesh Tewari ,"Blockchain-enabled End-to-End Encryption for Instant Messaging Applications", june 2022
[11] Hruthik B Gowda,Karun Datta Ramakumar,Sheethal.V,Sushma M,Dr. Madhusudhan G K "REAL-TIME VOICE CLONING USING DEEP LEARNING: A CASE STUDY"
[12] Fahima Khanam , Farha Akhter Munmun , Nadia Afrin Ritu, Aloke Kumar Saha , "Text To Speech Conversion Using Different Speech Synthesis"
[13] Dr.T NAnitha, Amilio Dsouza, Ashutosh , Akshay Gole ," REAL TIME VOICE CLONING"
[14] Tomáš Nekvinda, Ondˇrej Dušek, "One Model, Many Languages: Meta-learning for Multilingual Text-to-Speech"
[15] Dr. Abhay Kasetwar, Ritik Gajbhiye, Gopal Papewar, Rohan Nikhare, Priya Warade , "Development of chat application"
[16] Sakith Nalluri, A.Rohan Sai, and M.Saraswati, "Real Time Voice Cloning," vol. 7, pp. 297–302, April 2021
[17] Giuseppe Ruggiero, Enrico Zovato, Luigi Di Caro, Vincent Pollet, "Voice Cloning: A Multi-Speaker Text-To-Speech Synthesis Approach Based On Transfer Learning" , vol. 1 , Feb 2021.
[18] Anita and Srinivasan, "Text to Speech Conversion with Emotion Detection ," vol. 13, 14, 2018, pp. 11512-11517
[19] Noor Sabah, Jamal M. Kadhim and Ban N. Dhannoon, " Developing an End-to-End Secure Chat Application ", IJCSNS International Journal of Computer Science and Network Security, Vol.17 No.11, November 2017.
[20] Gonzalo Gómez Sánchez, " Voice Generation Using Deep Learning," September 28, 2016
[21] Sercan O. Arık, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, JonathanRaiman, and Yanqi Zhou, "Deep Voice 2: Multi-speaker neural text-to-speech". In NIPS, 2017b.
[22] Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A. C., and Bengio, Y, "SampleRNN: an unconditional end-to-end neural audio generation model". CoRR, abs/1612.07837, 2016.
[23] Emiliano Masi , Giovanni Cantone , Giuseppe Calavaro, "Mobile Apps Development: A Framework for Technology Decision Making", October 2012.
[24] R M Ali1, S N Alsaad1, "Instant messaging security and privacy secure instant messenger design", 2020.
[25] Jemine, Corentin," Automatic Multispeaker Voice Cloning",2019
[26] Hema A Murthy, "Building Speech Synthesis Systems for Indian Languages", 2017.
[27] Y. Lee, A. Rabiee, and S. Lee, "Emotional end-to-end neural speech synthesizer," CoRR, vol. abs/1711.05447, 2017.
[28] Masiello Eric, author, "Mastering React Native. January 11", 2017. Accessed 1 Jan 2022
[29] Naimul Islam Naim, ReactJS: An Open-Source JavaScript library for front-end development, Metropolia University of Applied Sciences, accessed on 1 Jan 2022