July 2023

A Type-to-Talk Framework Using Generative Voice Cloning Methods for Privacy-Preserving Communications

Dongeek Shin

A Type-to-Talk Framework Using Generative Voice Cloning Methods for Privacy-Preserving Communications

A conference platform allows users to connect through a video-based and/or an audio-based virtual meeting (e.g., a video conference or a conference call). A conference platform can enable audio-based and/or video-based conferences between multiple participants via respective client devices that are connected over a network and share each other's audio (e.g., voice of a user recorded via a microphone of a client device) and/or video streams (e.g., a video captured by a camera of a client device) during the conference.

In some instances, one (or more) of the participants may choose not to participate vocally in the meeting. As an example, a participant may be located in a public space, and may not want to disrupt other people in their surrounding environment who are not participating in the conference. For example, a participant may be located in a library or a busy airport, and may prefer not to participate in the conference vocally to avoid disrupting those around them. As another example, the topic of the meeting may be confidential, and the participant may want to avoid accidentally publicly disclosing sensitive information. In such instances, the participant is left with participating in the conference via text. For example, the participant may provide their input by typing in the chat feature of the conference platform, and the other participants can read the participant's input from the chat box in the user interface. This can lead to an asymmetric experience for all participants. A conference in which asymmetric communication modalities are used (i.e., in which one (or more) participant is communicating with the group via text (e.g., using the chat feature) and other participants are participating vocally) can break the flow of conversation and create an unnatural communication experience for everyone.

Aspects of the present disclosure address the above-noted and other deficiencies by providing a type-to-talk framework that generates speech mimicking a participant's voice during

a conference call. The framework can be implemented by the conference platform, and can use artificial intelligence to convert text (e.g., received from the chat feature) to speech. The framework can modify the generated speech to mimic the voice of the participant who provided the text. To implement the type-to-talk framework described herein, the platform can receive a speech sample for a participant. Using the speech sample, the framework can fine-tune a generative vocoder model for the specific participant, using the participant's vocal characteristics taken from the speech sample. Then, when the participant types text into the chat feature, the platform can use the customized generative vocoder to generate speech that sounds like participant's voice. The generated speech is then sent to the other participants, via a server hosting the conference. The type-to-talk framework described herein equalizes the conference experience by having each participant immersed in the conversation.

Figure 1 illustrates an example system architecture 100, which includes client devices 102A-N, a platform 120, and/or a server 130, each connected to a network 106. The network 106 may include a public network (e.g., the Internet), a private network (e.g., a local area network (LAN) or wide area network (WAN)), a wired network (e.g., Ethernet network), a wireless network (e.g., an 802.11 network or a Wi-Fi network), a cellular network (e.g., a Long Term Evolution (LTE) network), routers, hubs, switches, server computers, and/or a combination thereof. The platform 120 can be a video (or audio) conference platform, which can enable video-based (or audio-based) meetings between multiple participants via respective client devices 102A-N (e.g., that are connected over a network 106). A video conference refers to a real-time communication session such as a video conference call, also known as a video-based call or video chat, in which participants can connect with multiple additional participants in real-time and be provided with audio and video capabilities. Real-time communication refers to the

ability for users to communicate (e.g., exchange information) instantly without transmission delays and/or with negligible (e.g., milliseconds or microseconds) latency. The platform 120 can allow a user to join and participate in a video conference call with other users of the platform. The platform 120 can include a chat feature that enables participants to enter text that is shared with other participants of the conference.

The client devices 102A-N can each include computing devices such as personal computers (PCs), laptops, mobile phones, smart phones, tablet computers, netbook computers, network-connected televisions, etc. Each client device 102A-N can include an audiovisual component that can generate audio and video data to be streamed to video conference platform 120. The audiovisual component can include a device (e.g., a microphone) to capture an audio signal representing speech of a user and generate audio data (e.g., an audio file or audio stream) based on the captured audio signal. The audiovisual component can include another device (e.g., a speaker) to output audio data to a user associated with a particular client device 102A-N. The audiovisual component can also include an image capture device (e.g., a camera) to capture images and generate video data (e.g., a video stream) of the captured data of the captured images. Each client device 102A-N can include a platform application 110A-N, such as a web browser and/or a client application (e.g., a mobile application, a desktop application, etc.). The platform application 110A-N can present, on a display device of client devices 102A-N, a user interface for users to access platform 120. The user interface can include a chat box to enable users to enter and/or read text messages.

Server 130 can include a platform manager 122 that is configured to manage a conference (e.g., a virtual meeting) between multiple users of platform 120. Platform manager 122 can collect and provide data associated with the video conference to each participant of the video

conference. For example, an audiovisual component of each client device can capture images and generate video data (e.g., a video stream) of the captured data of the captured images. The client devices 102A-N can transmit the generated video stream to platform manager 122, and/or directly to other client devices 102A-N participating in the video conference. Additionally or alternatively, the audiovisual component of each client device can capture an audio signal representing speech of a user and generate audio data (e.g., an audio file or audio stream) based on the captured audio signal. The client devices 102A-N can transmit the generated audio data to platform manager 122, and/or directly to other client devices 102A-N.

Platform manager 122 can multiplex the video and/or audio data received from client devices 102A-N. Multiplexing can involve the integration and simultaneous transmission of multiple data streams, such as audio, video, chat messages, screen sharing, etc., over a single channel. The platform manager 122 can manage the receipt of various data streams from client devices 102A-N, process the data streams, and re-transmit the various data streams to the client devices 102A-N. The platform application 110A-N can receive the processed data streams from the platform manager 122, and can then provide the processed data stream to the user, e.g., via the audiovisual component of the client device 102A-N.

Platform application 110A-N can implement the type-to-talk framework described herein (note that in some instances, the type-to-talk framework described herein can be implemented by the platform manager 122). To implement the type-to-talk framework, platform application 110A can include an enrollment recording component 131, a speaker embeddings component 132, a generative vocoder component 133, and/or a generated speech component 134. Upon election to enable the type-to-talk framework (e.g., received as input from the user of client device 102A), the platform application 110A can initiate an enrollment process. Additionally or alternatively,

the enrollment process can be initiated via the settings of the platform application 110A, either once for each user that enables the type-to-talk framework, or once per instance of conference application 110A in which the user enables the type-to-talk framework. The enrollment process can include prompting the user to provide a speech sample. The prompt can provide a prepared script for the user to read. The prepared script can be designed to include all (or most) of the vowel and consonant sounds (e.g., including fricatives). The prepared script can provide a 10-second sample of the user's speech, for example. The enrollment recording component 131 can record the user's voice as they read the prepared script, and can optionally store the recording (i.e., the raw audio file) on client device 102A.

The speaker embeddings component 132 can analyze the recording of the speech sample. The analysis can be performed by a neural network, which can generate a hashed vector that represents the speaker embedding (i.e., a mathematical representation of a speaker's voice). For example, the speaker embeddings component 132 can preprocess the audio data by converting it into a spectrogram representing the distribution of the frequencies of the signal with respect to time. The neural network can be a recurrent neural network (RNN) that can include multiple layers. The neural network can be trained using a dataset of speech samples, each labeled with a corresponding vector representing a speaker's speech characteristics. The neural network can be trained to map each speech sample (or a representation of each speech sample) to the corresponding speaker's vector. Once trained, the neural network can then generate a speaker embedding vector for newly received audio speech samples. The generated speaker embedding can uniquely identify the speech patterns and vocal characteristics of a user of client device 102A. For example, the speaker embeddings component 132 can extract various features from the received audio, such as spectral content, fundamental frequency, and/or other acoustic

characteristics. Such features can capture the information needed to represent the speech signal. Implementing the type-to-talk framework on a client device 102A-N and thus storing the speaker embeddings locally (rather than transmitting the speaker embeddings via network 106) can provide privacy protection for the user of client device 102A-N.

The generative vocoder (or voice encoder) component 133 can be responsible to turning text into speech. The generative vocoder 133 can receive the user-specific speaker embedding from speaker embeddings component 132, and can store the received speaker embedding. The generative vocoder 133 can implement deep learning techniques such as recurrent neural networks (RNNs) or transformer models. The generative vocoder 133 can generate, based on the user's speaker embedding, new speech audio. For example, the generative vocoder 133 can create a synthesized waveform of the text received in the chat feature of the platform application 110A, and can apply a series of filters to this signal based on the user's specific speaker embedding.

As the user of client device 102A enters text into the chat feature of the platform application 110A, the generative vocoder 133 can generate speech audio of the text using the user's speaker embeddings. The generated audio thus mimics the user's vocal characteristics, and thus sounds like the user's voice. The generative vocoder 133 can generate the speech audio in real-time, as the user is entering text into the chat feature. For example, to simulate realistic speech timing, the generative vocoder 133 can include a real-time rendering engine that generates speech in a realistic timeframe. The rendering engine can run the generative vocoder 133 on a predetermined time schedule (e.g., every second as a user is entering text), and/or on a word-based timeframe (e.g., after every word is typed, or after every other word is typed). Rather than waiting for the user to finish typing, running the generative vocoder 133 periodically as the

user is entering text can result in a more natural representation of the user's speech, and can result in a more inclusive experience for the participants of the conference. The rendering engine can be optimized (e.g., based on each user's settings and/or typing habits) to determine a timeframe on which to run the generative vocoder 133 on the received text, in order to generate natural speech effects representing the user of client device 102A. Because the inference phase of the generated vocoder 133 runs frequently, the generative vocoder 133 is designed to be small and lightweight.

The generated speech component 134 can receive the speech generated by the generative vocoder 133 and combine multiple received speech signals from the generative vocoder 133. The generated speech component 134 can implement a semantic check over the multiple speech signals. For example, the semantic check can stitch together the last word of a received speech signal with the first word of the next received speech signal, in order to create a smooth speech representation of the user's text. By doing so, the generated speech component 134 can avoid sudden tonal switches or a discontinuities in pronunciations, for example. The generated speech component 134 can then send the generated speech to the server device 130 and/or to platform 120, via network 106. Platform manager 122 can receive audio from each client device 102A-N and performing the multiplexing described above. Platform manager 122 can then transmit the audio to each client device 102A-N participating in the conference.

The user interface of the platform application 110A-N can optionally display the text entered by the user of client device 102A (who is using the type-to-talk framework). The user interface of the platform application 110A-N can optionally display an indicator that indicates that the user of client device 102A is using the type-to-talk framework (i.e., that the speech is generated by artificial intelligence). For example, rather than displaying a audio symbol while

audio associated with client device 102A is being provided (i.e., while user of client device 102A is talking), the user interface of platform application 110A-N can display a text symbol while audio associated with client device 102A is provided (i.e., while the generated speech corresponding to the received text inputted into client device 102A is provided). Thus, the participants of the conference can be notified that the audio of the user of client device 102A is generated by artificial intelligence. As another example, displaying the text entered by the user of client device 102A while the audio of the text is being provided can function as the indicator that the speech is generated by artificial intelligence.

Abstract

A method for providing a text-to-speech framework that generated speech that mimics a user's voice is disclosed. The proposed method receives sample speech from the user, and generates speaker embeddings specific to the user. The speaker embeddings are generated using a neural network. The speaker embeddings are used to fine-tune a generative vocoder. The fine-tuned generative vocoder can be used to generate speech that mimics the speech patterns and vocal characteristics of the user. Thus, text entered by the user can be converted to audio that sounds like the user's speech. The generated audio is then transmitted to other participants in a virtual meeting.

Keywords: generative artificial intelligence, generative vocoder, generative voice encoder, text-to-speech, conference calls, virtual meetings, speaker embeddings, neural network, recurrent neural network

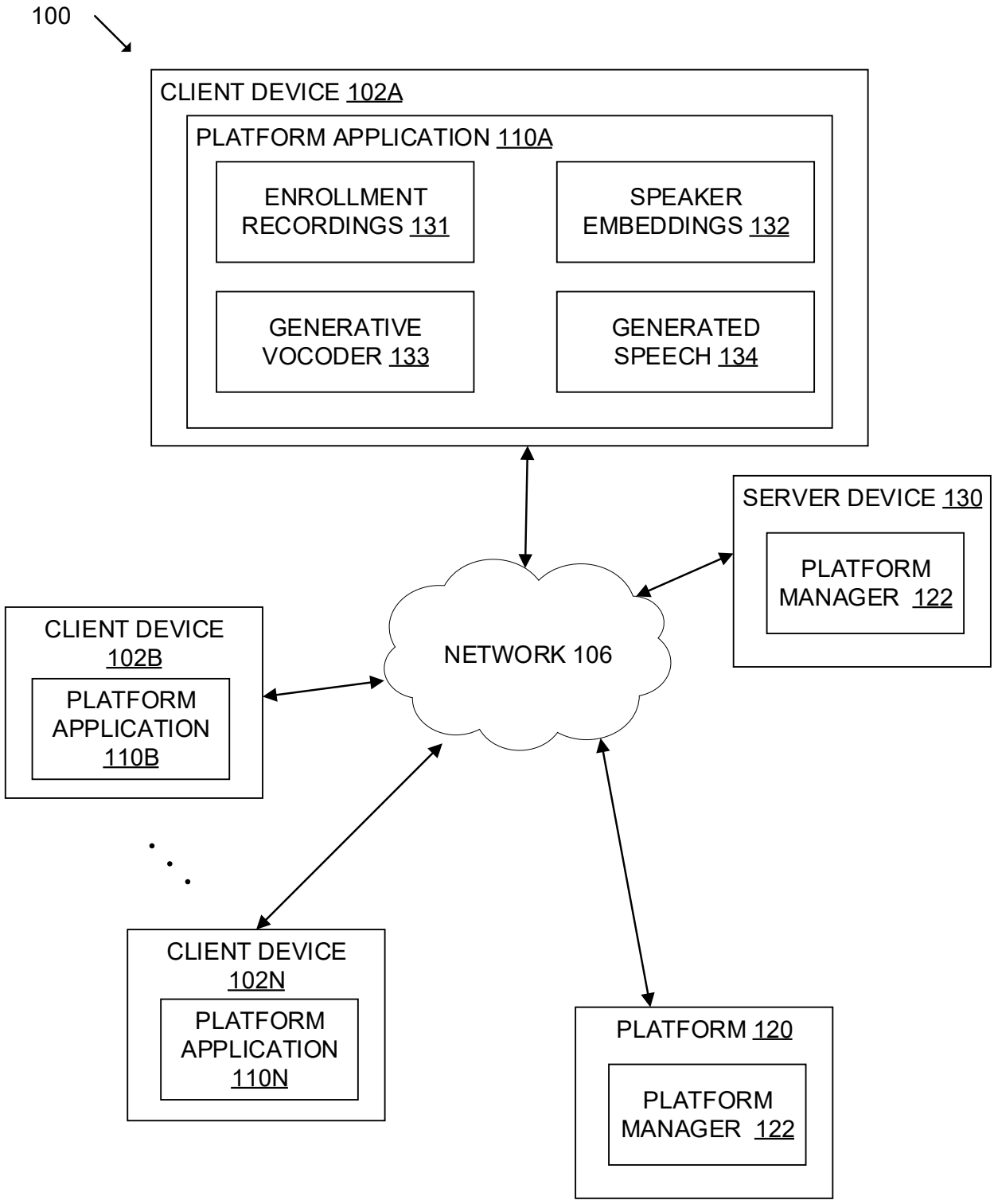
100
CLIENT DEVICE 102A
PLATFORM APPLICATION 110A
ENROLLMENT RECORDINGS 131
SPEAKER EMBEDDINGS 132
GENERATIVE VOCODER 133
GENERATED SPEECH 134
SERVER DEVICE 130
PLATFORM MANAGER 122
NETWORK 106
CLIENT DEVICE 102B
PLATFORM APPLICATION 110B
CLIENT DEVICE 102N
PLATFORM APPLICATION 110N
PLATFORM 120
PLATFORM MANAGER 122

Figure 1