Technical Disclosure Commons

Defensive Publications Series

June 2020

TRANSLATING BETWEEN REALTIME BI-DIRECTIONAL CONVERSATIONS AND PUSH-TO-TALK SYSTEMS

Faisal Siyavudeen

Sanjay Sinha

Ram Mohan R

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Siyavudeen, Faisal; Sinha, Sanjay; and R, Ram Mohan, "TRANSLATING BETWEEN REALTIME BI-DIRECTIONAL CONVERSATIONS AND PUSH-TO-TALK SYSTEMS", Technical Disclosure Commons, (June 15, 2020)
https://www.tdcommons.org/dpubs_series/3327

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

TRANSLATING BETWEEN REALTIME BI-DIRECTIONAL CONVERSATIONS AND PUSH-TO-TALK SYSTEMS

AUTHORS:

Faisal Siyavudeen
Sanjay Sinha
Ram Mohan R

ABSTRACT

The techniques presented herein provide an artificial intelligence (AI)-based conversational agent to allow communications between (i.e., "translate for") a push-to-talk (PTT) client and endpoints that support real-time, bi-directional conversation, including voice over Internet Protocol (VoIP) endpoints (also referred to as IP telephony endpoints) and endpoints running multimedia applications that allow bi-directional, real-time conversation. These techniques are applicable when PTT deployments allow direct calls between endpoints that support real-time, bi-directional conversation (e.g., VoIP endpoints) and PTT endpoints. During these calls, the techniques may allow a caller from the real-time, bi-directional conversation endpoint to have a real-time or nearly real-time conversational experience while the PTT user has a start-stop PTT experience.

DETAILED DESCRIPTION

Many push-to-talk systems allow integration with real-time, bi-directional conversation systems, such as VoIP communication systems (i.e., IP telephony systems), so that an end-to-end communication session can be established between a real-time, bi-directional conversation endpoints (for simplicity, referred to herein as a real-time conversation endpoint) and a PTT endpoint. However, typically user communication on such calls follow the PTT model, since the PTT side is often half-duplex and focuses on guaranteed delivery of messages. This PTT interaction may be fine for internal communication within a company, but the experience may be suboptimal for external callers calling into a PTT environment.

In the communications field, many AI agents (i.e., bots) are available; however, these bots do not provide a seamless, nearly real-time conversation experience for callers who are interacting with a PTT environment. That is, known bots do not appear to translate between a real-time conversation model and a PTT model. For example, some bots accept a call from a caller, sustain a conversation with the caller, collect information, dip into data sources, but then make calls to other humans to complete the transaction. Some bots may also add conversational niceties around data. Alternatively, other AI agents may convert speech to text, pass the text to natural-language understanding processes, and respond based on determined intents. However, these bots do not facilitate a real-time conversation for a caller calling a PTT system and expecting a full duplex experience. That is, these bots do not bridge a human calling from a real-time conversation endpoint (e.g., a VoIP endpoint) with a human in a PTT system.

The techniques presented herein bridge this gap by providing an AI-based conversational agent that allows communications between a PTT client and a real-time conversation endpoint (e.g., a VoIP / IP telephony endpoint). The AI-based agent (i.e., an AI-based bot) can retain a stop-start experience on the PTT side while providing basic conversation semantics to fill in the gaps for a caller (from the real-time conversation endpoint) such that it seems like a conversation to the caller. Thus, a caller calling into a PTT system implementing these techniques may know that he/she is talking to an automated assistant, but the caller would be able to converse with a human who is working within a PTT system outside the constraints of a pure PTT conversation.

For example, if a retail store is implementing a PTT system, a caller calling the retail store will be routed to a specific channel, such as a channel dedicated to its bakery, and any PTT user in that channel may take the call. Then, a potential conversation with the PTT user facilitated by the techniques presented herein may flow as follows (with “AI Agent” representing the techniques presented herein):

Caller: “Hi, would you have cakes in stock?”

AI Agent: “Hello, thanks for calling us. Can you please wait while I check?”

AI Agent->PTT (Bakery): “Hi, would you have cakes in stock”

PTT User->AI: “What type of cake are you looking for?”

AI-Agent->Caller: “Thanks for waiting. What type of cake are you looking for?”

- and so on.

Notably, during this call flow, the AI agent will play the caller's audio on the PTT channel. The channel could correspond to a single PTT user or a PTT group. Meanwhile, the audio from the PTT channel can be analyzed to determine how to process the audio before playing audio out to the caller.

Specifically, the audio could be analyzed to determine how much the accent, pitch, and tone of the PTT user's audio vary from the AI agent's "voice." If the accent, pitch, and tone do not vary too much from to the AI agent's voice, the pitch and tone could be modulated to match the AI agent voice and the modulated audio can be passed to the caller. In these scenarios, the AI agent is essentially a translator between full duplex and half duplex conversation modes. However, if there is too much variation, the PTT user's audio can be converted to text and then re-converted to audio that uses the AI agent's "voice."

In some cases, the AI agent can be a conversational bot with a generic model that allows it to fill in conversational niceties around the actual information during a conversation (e.g., prior to, subsequent to, and/or while transmitting audio from the PTT channel). This would allow the agent to be deployed with minimal training. For example, if the AI agent is acting as a translator between full duplex and half duplex conversation modes (i.e., the first option mentioned above), the AI agent may combine PTT audio with conversational fillers before passing the audio to the caller.

Alternatively, the AI agent can be trained on domain-specific dialog models to offer a better conversational experience prior to, subsequent to, and/or while transmitting audio from the PTT channel. In some instances, the content of the conversations, including the responses from the PTT side on the current conversation or the previous conversations, can be used as source material to adapt the responses generated by the AI agent. However, to analyze responses as source material, the AI agent may need to convert speech to text (e.g., in accordance with the second option for passing PTT audio to a caller discussed above).

Either way, in at least some instances, the AI agent can be configured to pick up and play information from some data sources (e.g., enterprise resource planning (ERP) systems, sensor systems from industrial automation, etc.) so that the AI agent can respond to some caller inquiries without asking questions on a PTT channel. This information (e.g., store menu, availability information, etc.) may need to be curated to prevent internal information from being exposed. Additionally, if the information provided by these data sources differs from the responses provided from a PTT human user, the AI agent may need to determine whether to use the information from the data sources. As one example, in these scenarios, a relevance score (based on freshness of content, topical relevance, etc.) can be assigned to the recommendation from the data sources to determine whether to use the information from the data sources. If the relevance score is low, one or more PTT interactions may be used to confirm that information should be used before a response is framed for play-out on the caller side. However, again, to analyze responses, the AI agent may need to convert speech to text (e.g., in accordance with the second option for passing PTT audio to a caller discussed above)

An example high-level solution architecture for the techniques presented herein is illustrated in Figure 1 below.

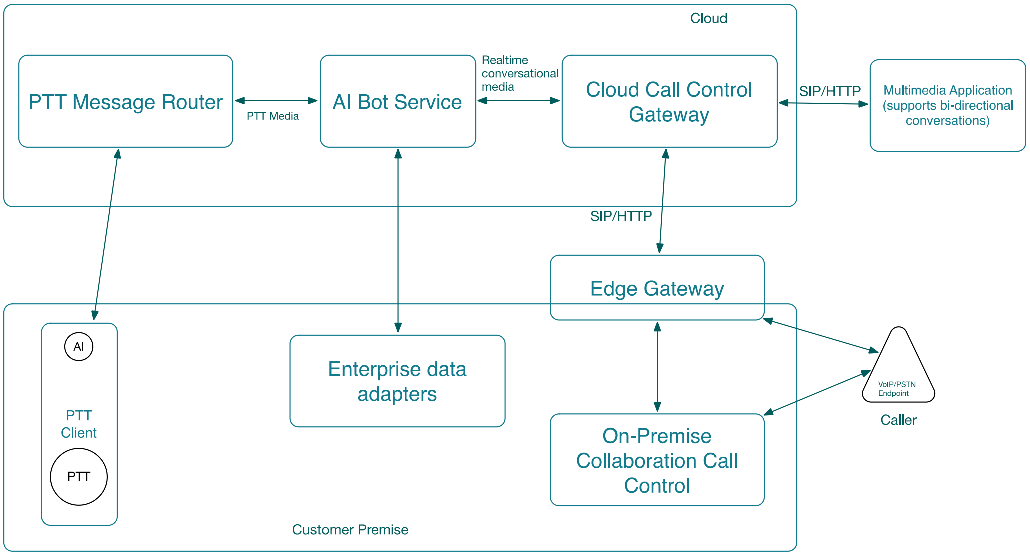
Cloud
PTT Message Router
PTT Media
AI Bot Service
Realtime conversational media
Cloud Call Control Gateway
SIP/HTTP
Multimedia Application (supports bi-directional conversations)
SIP/HTTP
Edge Gateway
AI
PTT Client
PTT
Enterprise data adapters
On-Premise Collaboration Call Control
VoIP/PSTN Endpoint
Caller
Customer Premise

Figure 1

However, Figure 1 is just an example and, in at least some instances, the techniques may also be utilized with systems that pass context of the caller to the AI agent. For example, in a Web Real-Time Communication (WebRTC) click-to-call scenario where a user is browsing a web store and clicks to call a specific store, the browsing behavior of the user can be sent over the call to the AI agent. Then, the AI agent can use this browsing behavior to augment the conversation and select the proper PTT channel.

As an example, the Java Script running on a browser downloaded from a store site when a user triggers to make call could create a one-time token (OTT) that references browsing context. Alternatively, a call could transit from WebRTC to Session Initiation Protocol and the OTT token can be carried in a SIP header. Either way, the call control infrastructure that handles a call views the OTT as an opaque object. Then, the AI agent service will hand the OTT token to the enterprise data adapter, which will inspect the OTT to fetch the context and select the appropriate PTT group to which this call should be bridged, based on the context. The AI agent can use this information to bridge the call between caller and the proper PTT group.

Overall, the techniques presented herein may not alter how voice content is passed from a caller (e.g., from a real-time conversation endpoint) to the PTT side, but the techniques may improve the experience for the caller by improving the PTT side. For example, in the simplest cases, the AI agent may provide communications that are nearly cut-through audio communications (with some accent and voice modulation processing) from the PTT side to the caller, creating a conversational experience for a caller. If, however, the techniques perform multiple interactions (e.g., as described above) that cause a delay in audio playback to the caller that is longer than a predefined threshold, the calling user can be put on hold or the AI agent can fill the time with general conversational content.

In sum, the techniques described herein provide an AI-based conversational agent (i.e., a bot) to allow communications between (i.e., "translate for") a PTT client and endpoints that support real-time, bi-directional conversation, including VoIP endpoints and endpoints running multimedia applications that allow bi-directional, real-time conversation. These techniques are applicable when PTT deployments allow direct calls between endpoints that support real-time, bi-directional conversation (e.g., VoIP endpoints) and PTT endpoints. During these calls, the techniques may allow a caller from the real-time, bi-directional conversation endpoint to have a real-time or nearly real-time conversational experience while the PTT user has a start-stop PTT experience.