January 23, 2019

A Virtual Call Assistant

Anonymous

A Virtual Call Assistant

Abstract

A system for adding a virtual assistant as a user to a call is disclosed. The virtual assistant (implemented as a web Real Time Communication (webRTC) client) can be added by a manual selection on a Graphical User Interface (GUI) of a calling application (app) when a call is in progress between users. The virtual assistant responds to queries of the users. The users may ask queries from the virtual assistant using a trigger phrase, such as “Hi assistant!”. The virtual assistant implements speech recognition algorithms and searches for requisite information in a knowledge graph over the Internet and answers explicit queries by talking back (voice response) to the users. The answer can also be provided as a text or a video. For implicit queries, the virtual assistant renders information as data channel packets over a webRTC data channel. The rendered information may be in the form of video or text.

Problem

While on an ongoing conversation call, users of the call might have doubts/queries regarding an agenda that they are talking about. They might want to get assistance from an online search engine. But, they will have to toggle between browsers to search content to resolve the doubts/queries. This might be distracting for the users as this will take away their attention from the ongoing conversation. In case the users wished to resolve their queries after the conversation, it is a possibility that they might not remember their queries once the conversation is concluded. An effort to constantly noting down queries during the conversation will also distract them.

The present disclosure endeavors to solve these problems by providing a real time assistance for resolving the doubts/queries from the users in the conversation using a virtual assistant.

Description

System

The system described in the present disclosure is a calling application (app) that provides an option of adding a virtual assistant to a call (between a first user and a second user) as shown in Figure 1.

The calling app additionally comprises of following components:

1. A VXML (Voice Extensible Markup Language) engine.
2. A server
3. A knowledge graph
4. A Graphical User Interface (GUI)

The virtual assistant further includes:

i. A speech processing engine
ii. A text to speech converter
iii. A rendering unit

The speech processing engine implements speech to text conversion algorithms. The VXML engine enables the virtual assistant to be added to the call as a user when either of the first user or the second user makes a manual selection on the GUI of the app to add the virtual assistant. The virtual assistant is implemented as a web Real Time Communication (webRTC) client. The server of the app includes a database that stores pronunciations of dictionary words. The virtual assistant searches for information relevant to a set of keywords (detected in a query) in the knowledge graph over the Internet. The knowledge graph is a knowledge base that integrates information acquired from a variety of sources and provides improved search results. The rendering unit interprets an HTML code of a webpage delivered by the knowledge graph for making the webpage understandable by the user.

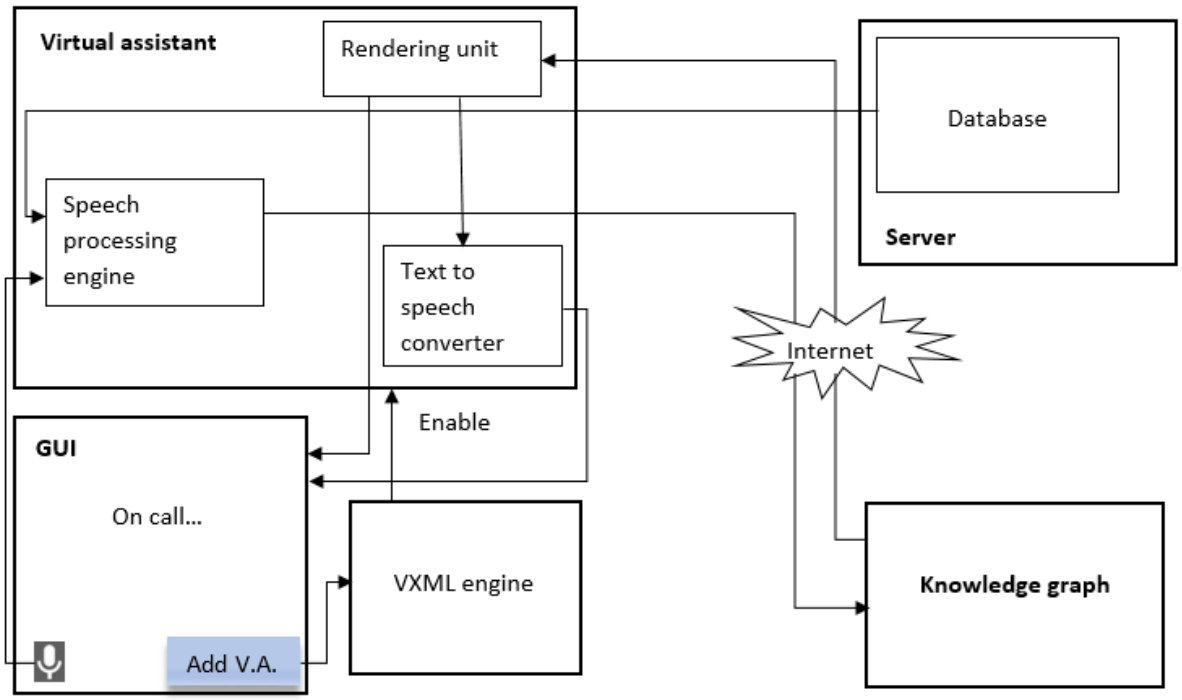

Figure 1: System architecture of the calling app for adding the Virtual Assistant (V.A.) to the call

In one embodiment, the virtual assistant is capable of understanding and responding to implicit queries as well, by employing a dialog-state system. The dialog-state system, as demonstrated in Figure 2, comprises of the following units:

- An Automatic Speech Recognition (ASR) Unit
- A Natural Language Understanding (NLU) Unit
- A Dialog-State Tracker (DST) Unit
- A Dialog Policy Unit
- A Rendering Unit

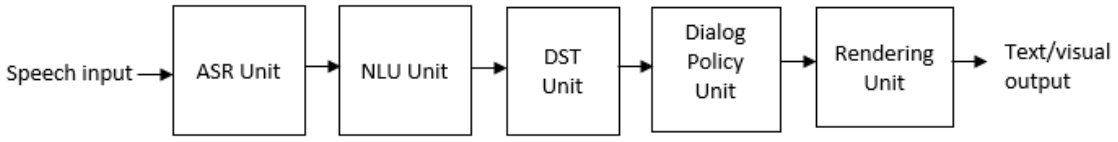

Figure 2: A block diagram of the dialog state system

Working

When the first user calls the second user using the calling app, a VXML script in the VXML engine is triggered. The VXML script enables adding the virtual assistant to the call. The GUI provides an option for manual selection that helps to add the virtual assistant to the call on a device of each user. A VXML event is triggered when either of the users makes the manual selection. The VXML event executes functions in the VXML script for enabling the virtual assistant. The virtual assistant is added as a user in the call as a response to the VXML event as shown in the Figure 3.

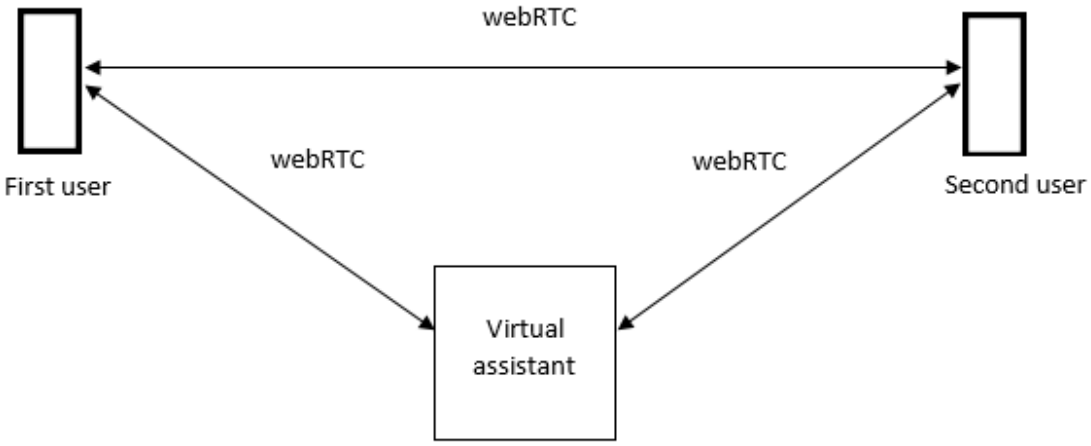

Figure 3: A three-way group call between the users and the virtual assistant

When one of the users asks a query from the virtual assistant, the latter checks for a trigger phrase in the query. The trigger phrase may be “Hi assistant!”. If, the trigger phrase is not detected, the virtual assistant discards the query (i.e., the virtual assistant does not respond). Once a query starting with the trigger phrase is detected, the virtual assistant denormalizes the query. The denormalization removes the trigger phrase from the query. For example, a query “Hi assistant! What’s the time?” is denormalized to “What’s the time?”.

The virtual assistant records the user’s query in real time at the server of the calling app. The recorded query is segmented into a plurality of sound fragments. The virtual assistant searches for a set of pronunciations in the database that matches each of the sound fragments. The resultant set of pronunciations is converted into a set of keywords using speech to text conversion algorithms. The virtual assistant searches for information relevant to the identified keywords in the knowledge graph over the

Internet and the information retrieved is transmitted back to the user's (who had initiated the query) device. The information may be in the form of text, audio or video. If the virtual assistant needs to talk back, the identified keywords are transformed into speech using text to speech conversion algorithms.

Additional embodiments

In an embodiment, the virtual assistant may keep listening to the words uttered by the users continuously. The virtual assistant checks if any word(s) is/are uttered repeatedly during the conversation. The sound corresponding to each repeated word is matched for similarity with the set of pronunciations of words stored in the database. A best possible matched pronunciation is converted into a keyword using speech to text conversion algorithms. The keyword is searched in a repository, which may be the Internet. The virtual assistant hyperlinks most relevant information gathered from the repository. The hyperlinked information is presented on the GUI of the device of each user. For example, if there is a repeated mention of "autonomous vehicles" in the conversation, a hyperlink to a webpage that describes working of the autonomous vehicles is created. When one of the users accesses the hyperlink, the webpage corresponding to the working of the autonomous vehicles is rendered separately over the webRTC channel.

In one of the embodiments, the virtual assistant is capable of understanding and responding to implicit queries as well. For example, the first user might ask the second user that how likely is it for a particular batsman to score a ton at a venue. In response to the said implicit query, the virtual assistant surfaces a page that shows the batsman's record at the venue. The virtual assistant utilizes the dialog-state system for interpreting and answering the implicit queries. The dialog-state system determines a dialog-act, which the first user is using. For example, the dialog state system interprets whether the first user is asking a question from the second user. The dialog act corresponding to asking the question is called a directive. The ASR unit recognizes a speech input given by the first user. The NLU unit produces a semantic representation from the speech input. The dialog-state tracker unit determines the dialog act from the speech input. The dialog policy unit decides whether to address the dialog act. If the dialog act is a question/request, the rendering unit loads a text/visual content related to the implicit query on the device(s) of the user(s).

The virtual assistant can be further extended to pick up action items from the ongoing conversation. It can take notes for follow-ups from the conversation. This eases up the burden on the users of

remembering every important aspect covered in the conversation. Further, the virtual assistant is capable to set up reminders on the device(s) of the users in the conversation for future action items.

Conclusion

Discussions/conversations might often raise some doubts/queries amongst users about an agenda that is being talked about. The users would require assistance on resolution of their doubts/queries. Also, it is likely that additional details about the topics are helpful to take the conversation forward. The virtual assistant, as described in the present disclosure, acts as a supportive application that can be added as a user to the ongoing discussion/conversation to provide the users with the above real-time services.