CUSTOMIZED VIRTUAL AGENT SYSTEM

BACKGROUND

[0001] An online marketplace may provide a number of services, such as accommodations, tours, transportation and the like, and allow users to reserve or "book" one or more services. For example, a first user (e.g., host) can list one or more services on the online marketplace and a second user (e.g., guest) can request to view listings of services for a particular location (e.g., San Francisco) that may include a listing for the first user's service. Current customer support systems in such online marketplaces face significant inefficiencies.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Various ones of the appended drawings merely illustrate example embodiments of the present disclosure and should not be considered as limiting its scope.

[0003] FIG. 1 is a block diagram illustrating a networked system, according to some examples.

[0004] FIG. 2 is a block diagram illustrating example components of a customized virtual agent system, according to some examples.

[0005] FIG. 3 is a block diagram illustrating a reservation system, according to some examples.

[0006] FIG. 4 is a flow chart illustrating aspects of a method, according to some examples.

[0007] FIG. 5 is a flow chart illustrating aspects of a method, according to some examples.

[0008] FIG. 6 is a flow chart illustrating aspects of a method, according to some examples.

[0009] FIG. 7 diagrammatically illustrates a machine learning pipeline, according to some examples.

[0010] FIG. 8 is a block diagram illustrating an example of a software architecture that may be installed on a machine, according to some examples.

[0011] FIG. 9 illustrates a diagrammatic representation of a machine, in the form of a computer system, within which a set of instructions may be executed for causing the machine to perform any one or more of the methodologies discussed herein, according to some examples.

DETAILED DESCRIPTION

[0012] Systems and methods described herein relate to a customized virtual agent system. Current customer support systems in such online marketplaces face significant inefficiencies, such as inefficient outbound calling and voicemail handling, language barriers, and lack of integration into booking services. For example, customer support agents spend considerable time making outbound calls to users to resolve issues between guests and hosts. A large percentage of these calls end in voicemail, requiring agents to manually listen to greetings before leaving messages, resulting in wasted time and reduced operational efficiency. This is particularly problematic in two-sided marketplaces (e.g., with hosts and guests) where mediation between parties is frequently required.

[0013] Moreover, international travelers often require support in languages that they do not speak, creating communication barriers between guests, hosts and customer support agents. Current solutions rely on human translators joining a conference call. This is not only costly, but is dependent on translator availability real-time, which is often lacking. Accordingly, this is time consuming and can delay issue resolution. This is particularly significant in a situation where a user may have an urgent or emergency situation and no translator is available. Further, a third-party interpreter must join in on sensitive conversations, introducing privacy concerns.

[0014] In addition, when a host of a listing in an online marketplace cancels a reservation shortly before check-in, it is not possible for a human agent, in real-time, to call the effected user(s), search for similar available accommodations, apply policy-based compensation and arrange and book alternative listings, among other needed tasks in this scenario. This is further compounded when a user is not available and a call goes to voicemail or translations services are needed, as explained above.

[0015] To address these and other technical problems, the customized virtual agent system is described herein utilizes artificial intelligence, such as by using one or more large language models (LLMs), and an interactive voice response (IVR) system that autonomously places outbound calls triggered by customer support workflows and detects whether a live person or a voicemail system answers. The customized virtual agent system can automatically handle voicemail interactions without a customer support agent involvement significantly reducing time and operational costs. Moreover, the customized virtual agent system provides real-time translations between customer support agents and users with customized translations corresponding to a user locale and language preferences. Accordingly, the customized virtual agent system can provide real-time translations exactly when needed, matching user accents and dialects while maintaining privacy within the online marketplace. Further, the customized virtual agent system can autonomously detect a last-minute cancellation, identify similar available listings to the canceled listing, apply dynamic pricing adjustments based on the canceled listing, book an alternate listing and even call a user to confirm details for booking the alternate listing, in real-time upon detecting a last-minute cancelation. Further details of the customized virtual agent system are described below.

[0016] FIG. 1 is a block diagram illustrating a networked system 100, according to some example embodiments. The networked system 100 can include one or more computing devices such as a client device 110. The client device 110 may comprise, but is not limited to a mobile phone, desktop computer, laptop, portable digital assistant (PDA), smart phone, tablet, ultrabook, netbook, laptop, multiprocessor system, microprocessor-based or programmable consumer electronic system, game console, set-top box, computer in a vehicle, wearable device (e.g., smart watch, smart glasses), or any other communication device that a user may utilize to access the networked system 100. In some embodiments, the client device 110 comprises a display component (not shown) to display information (e.g., in the form of user interfaces). In further embodiments, the client device 110 comprises one or more of touch screens, accelerometers, gyroscopes, cameras, microphones, Global Positioning System

(GPS) devices, and so forth. The client device 110 may be a device of a user (e.g., guest) that is used to request and receive reservation information, accommodation information, and entry and access information for a reserved accommodation; set or update user preferences; request to view listings; view listings results in a list form or in a maps viewport; connect with a host; review details for a canceled reservation; review options for one or more alternate listings; contact a customer support agent (e.g., call or message); and so forth, associated with travel or other products or services in an online marketplace. The client device 110 may also be a device of a user (e.g., host) that is used to post and maintain a listing for an accommodation, experience or service; request and receive reservation information and guest information; generate entry and access information (e.g., access codes); set or update user preferences; request for and view options for co-hosts; cancel a reservation; and so forth. The client device 110 may further be a device of a user (e.g., customer support agent) that is used to contact a user (e.g., call or message a guest or host), review reservation cancelation details, review and select from one or more alternate listings for a canceled reservation, and so forth.

[0017] One or more users 106 may be a person (e.g., guest, host, service personnel, customer support agent), a machine, or other means of interacting with the client device 110. In example embodiments, the user 106 may not be part of the networked system 100 but may interact with the networked system 100 via the client device 110 or other means. For instance, the user 106 can provide input (e.g., voice input, touch screen input, alphanumeric input) to the client device 110 and the input may be communicated to other entities in the networked system 100 (e.g., third-party servers 130, a server system 102) via a network 104. In this instance, the other entities in the networked system 100, in response to receiving the input from the user 106, can communicate information to the client device 110 via the network 104 to be presented to the user 106. In this way, the user 106 can interact with the various entities in the networked system 100 using the client device 110.

[0018] The networked system 100 further includes a network 104. One or more portions of the network 104 can be an ad hoc network, an intranet, an extranet, a

virtual private network (VPN), a local area network (LAN), a wireless LAN (WLAN), a wide area network (WAN), a wireless WAN (WWAN), a metropolitan area network (MAN), a portion of the internet, a portion of the public switched telephone network (PSTN), a cellular telephone network, a wireless network, a Wi-Fi network, a WiMAX network, another type of network, or a combination of two or more such networks.

[0019] One or more portions of the network 104 can comprise short-range wireless communication, such as Bluetooth, WiFi, near field communication (NFC), ultraband, Zigbee, or other form of short-range wireless communication.

[0020] The client device 110 can access the various data and applications provided by other entities in the networked system 100 via a web client 112 (e.g., a browser, such as the Internet Explorer® browser developed by Microsoft® Corporation of Redmond, Washington) or one or more client applications 114. The client device 110 can include one or more client applications 114 (also referred to as "apps") such as, but not limited to, a web browser, a messaging application, an electronic mail (email) application, an ecommerce site application, a mapping or location application, a reservation application, an entry or keypad access application, a customer support application, and the like.

[0021] In some embodiments, one or more client applications 114 can be included in a given one of the client devices 110 and configured to locally provide the user interface and at least some of the functionalities, with the client application 114 configured to communicate with other entities in the networked system 100 (e.g., third-party servers 130, the server system 102), on an as-needed basis, for data and/or processing capabilities not locally available (e.g., to access reservation or listing information, request data, authenticate a user 106, verify a method of payment, receive an access code). Conversely, one or more client applications 114 may not be included in the client device 110, and then the client device 110 can use its web browser to access the one or more applications hosted on other entities in the networked system 100 (e.g., third-party servers 130, the server system 102).

[0022] The networked system 100 can further include one or more third-party

servers 130. The one or more third-party servers 130 may include one or more third-party application(s) 132. The one or more third-party application(s) 132, executing on the third-party server(s) 130, can interact with the server system 102 via a programmatic interface provided by an application programming interface (API) gateway server 120. For example, one or more of the third-party applications 132 can request and utilize information from the server system 102 via the API gateway server 120 to support one or more features or functions on a website hosted by a third party or an application hosted by the third party. The third-party website or application 132, for example, can provide various functionality that is supported by relevant functionality and data in the server system 102, such as entry and access information for an accommodation. The third-party servers 130 can be a cloud computing environment, according to some example embodiments. The third-party servers 130, and any servers associated with the third-party servers 130, can be associated with a cloud-based application, in one example embodiment.

[0023] The server system 102 can provide server-side functionality via the network 104 (e.g., the internet or a WAN) to one or more third-party servers 130 and/or one or more client devices 110 and/or one or more accommodation devices 140. The server system 102 is a cloud computing environment, according to some example embodiments. The server system 102, and any servers associated with the server system 102, are associated with a cloud-based application, in one example embodiment.

[0024] In one example, the server system 102 provides server-side functionality for an online marketplace. The online marketplace provides various listings for trip items, such as accommodations hosted by various managers (also referred to as “owners” or “hosts”) that can be reserved by clients (also referred to as “users” or “guests”), such as an apartment, a house, a cabin, one or more rooms in an apartment or house, and the like. The online marketplace can provide for methods for hosts to search for and connect with co-hosts to manage one or more listings in the online marketplace. The online marketplace can further provide listings for other trip items, such as experiences (e.g., local tours), car rentals, flights, public transportation, and other transportation or activities related to

travel.

[0025] The server system 102 can include the API gateway server 120, a web server 122, a reservation system 124 and a customized virtual agent system 128 that can be communicatively coupled with one or more databases 126 or other forms of data store.

[0026] The customized virtual agent system 128 provides various functionality for communicating with users in an online marketplace, as mentioned above. FIG. 2 is a block diagram illustrating example components of the customized virtual agent system 128, such as IVR core services 202, an LLM orchestrator 204, a translation virtual agent 206, a rebooking virtual agent 208, and a voicemail detection virtual agent 210. It is to be understood that the customized virtual agent system 128 can comprise more, less, or different components in examples discussed herein.

[0027] The IVR core services 202 is configured as a communication interface between users and the customized virtual agent system 128. For example, the IVR core services 202 can receive inbound calls and route calls to customer support agents. The IVR cores services 202 can handle speech-to-text (SST) and text-to-speech (TTS) conversion for all voice communications. The SST and TTS capabilities can be used to communicate with one or more LLMs that take text input and provide text output. The IVR core services 202 can further facilitate handovers between virtual agents and customer support agents when needed and are capable of handling multi-part conversations, such as between guests, hosts, and customer support agents.

[0028] The LLM orchestrator 204 is configured as a reasoning engine to determine which specialized virtual agent is most suitable for each customer support task and can evaluate scenarios and decide whether customer support agent involvement is required. For example, the LLM orchestrator 204 can route a customer support task to the translation virtual agent 206, the rebooking virtual agent 208, the voicemail detection virtual agent 210, or other virtual agent. The LLM orchestrator 204 can further generate prompts for an LLM based on a specific customer support task and virtual agent and coordinate multiple virtual agents working together.

[0029] The translation virtual agent 206 is configured as an LLM, or the like, that can join a call with a customer support agent and one or more users and can process real-time translation and generate responses in an appropriate language. In one example, the LLM orchestrator 204 generates a dynamic prompt with instructions for the translation virtual agent 206 to handle a call where language translation is needed. In some examples, the LLM utilizes STT to translate spoken output from each party and uses TTS to deliver the translated speech back to the relevant party. In other examples, the LLM can process audio directly to generate a translation of the audio and provide the translated audio back to the relevant party.

[0030] In some examples, the translation virtual agent 206 is configured to detect a user locale and language preference, detect specific accents and regional language variations, can adjust voice output based on terminology specific to a given online marketplace, and the like. The translation virtual agent 206 can work with other virtual agents such as the rebooking agent 208 and the voicemail detection virtual agent 210 when language translation is needed by other virtual agents.

[0031] The rebooking virtual agent 208 can be triggered when a last-minute cancelation is detected. The rebooking virtual agent 208 can also be configured as an LLM, or the like. The rebooking virtual agent 208 can call a user immediately to assist with finding alternative accommodations, leverage one or more machine learning models to determine one or more similar listings, and present top similar listings to a computing device of the affected user(s). The rebooking virtual agent 208 can handle dynamic pricing by calculating and offering policy-based discounts for price differences between a canceled reservation and alternate listings for rebooking. The rebooking virtual agent 208 can escalate a call to a customer support agent for out-of-policy requests. The rebooking virtual agent 208 can work with the translation virtual agent 206 for multilingual rebooking scenarios and can coordinate outbound call handling and/or the voicemail detection virtual agent 210 for outbound calls. In some examples, the rebooking virtual agent 208 is an LLM that is fine-tuned based on training data and customer service conversations within a given online

marketplace to converse with a user in an appropriate tone and empathy for urgent situations.

[0032] The voicemail detection virtual agent 210 is configured as an LLM, or the like, that can automatically place an outbound call and detect if a live person answers or if a voicemail is reached. In some examples, the LLM utilizes STT to convert voice input to text to determine whether the voice input is a live person or voicemail. In other examples, the LLM can process the voice input directly without a STT conversion.

[0033] The virtual agents 206-210 are LLMs that are a form of generative artificial intelligence, also known and generative AI or GenAI. Generative AI is a type of artificial intelligence that utilizes machine learning techniques to generate new data based on training data. The new data can comprise text, images or audio. Some examples of LLMs include DeepSeek, Alibaba, Llama, Mixtral, Flan-T5, and the like, that are initially trained on massive datasets to learn general language understanding and patterns. For the customized agent system 128, one or more of these LLMs are fine-tuned by further training the pre-trained LLM on a focused dataset comprising data specific to the online marketplace and a specific task, such as language translation, rebooking, voicemail detection, or other tasks of the online marketplace. Accordingly, each LLM described herein is a specialized (customized) LLM trained for a specific task.

[0034] Further details about the functions of the customized virtual agent system 128 are described with respect to FIGS. 4-6.

[0035] Returning to FIG. 1, the one or more databases 126 can be one or more storage devices that store data related to the reservation system 124, the customized virtual agent system 128, and other systems or data. The one or more databases 126 can further store information related to third-party servers 130, third-party applications 132, client devices 110, client applications 114, users 106, and so forth. The one or more databases 126 can be implemented using any suitable database management system such as MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SAP, IBM DB2, or the like. The one or more databases 126 include cloud-based storage in some embodiments.

[0036] The reservation system 124 manages resources and provides back-end support for third-party servers 130, third-party applications 132, client applications 114, and so forth, which may include cloud-based applications. The reservation system 124 provides functionality for viewing listings related to trip items (e.g., accommodation listings, activity listings), generating and posting a new listing, analyzing and ranking images to be posted in a new listing, managing listings, booking listings and other reservation functionality, and so forth, for an online marketplace. Further details related to the reservation system 124 are shown in FIG. 2.

[0037] FIG. 3 is a block diagram illustrating a reservation system 124, according to some example embodiments. The reservation system 124 comprises a front-end server 302, a client component 304, a manager component 306, a listing component 308, a search component 310, and a transaction component 312. The one or more database(s) 126 include a client store 314, a manager store 316, a listing store 318, a query store 320, a transaction store 322, and a booking session store 324. The reservation system 124 may also contain different and/or other components that are not described herein.

[0038] The reservation system 124 can be implemented using a single computing device or a network of computing devices, including cloud-based computer implementations. The computing devices can be server-class computers including one or more high-performance computer processors and random access memory, which can run an operating system such as Linux or the like. The operations of the reservation system 124 can be controlled either through hardware or through computer programs installed in nontransitory computer-readable storage devices such as solid-state devices or magnetic storage devices and executed by the processors to perform the functions described herein.

[0039] The front-end server 302 includes program code that allows client devices 110 to communicate with the reservation system 124. The front-end server 302 can utilize the API gateway server 120 and/or the web server 122 shown in FIG. 1. The front-end server 302 can include a web server hosting one or more websites accessible via a hypertext transfer protocol (HTTP), such that

user agents, such as a web browser software application, can be installed on the client devices 110 and can send commands to and receive data from the reservation system 124. The front-end server 302 can also utilize the API gateway server 120 that allows software applications installed on client devices 110 and third-party servers 130 and applications 132 to call to the API to send commands to and receive data from the reservation system 124. The front-end server 302 further includes program code to route commands and data to the other components of the reservation system 124 to carry out the processes described herein and respond to the client devices 110 accordingly.

[0040] The client component 304 comprises program code that allows clients (also referred to herein as "users" or "guests") to manage their interactions with the reservation system 124 and executes processing logic for client-related information that may be requested by other components of the reservation system 124. Each client is represented in the reservation system 124 by an individual client object having a unique client identifier (ID) and client profile, both of which are stored in the client store 314.

[0041] The client profile includes a number of client-related attribute fields that can include a profile picture and/or other identifying information, a geographical location, a client calendar, an access code, smart device preferences (e.g., user preferences), and so forth. The client's geographical location is either the client's current location (e.g., based on information provided by the client device 110) or the client's manually entered home address, neighborhood, city, state, or country of residence. The client location may be used to filter search criteria for time-expiring inventory relevant to a particular client or to assign default language preferences.

[0042] The client component 304 comprises program code to provide for clients to set up and modify the client profile. The reservation system 124 allows each client to exchange communications, request transactions, and perform transactions with one or more managers.

[0043] The manager component 306 comprises program code that provides a user interface that allows managers (also referred to herein as "users," "hosts," "co-hosts" or "owners") to manage their interactions and listings with the

reservation system 124 and executes processing logic for manager-related information that may be requested by other components of the reservation system 124. Each manager is represented in the reservation system 124 by an individual manager object having a unique manager ID and manager profile, both of which are stored in the manager store 316.

[0044] The manager profile is associated with one or more listings owned or managed by the manager and includes a number of manager attributes including transaction requests and a set of listing calendars for each of the listings managed by the manager.

[0045] The manager component 306 provides code for managers to set up and modify the manager profile listings. A user 106 of the reservation system 124 can be both a manager and a client. In this case, the user 106 will have a profile entry in both the client store 314 and the manager store 316 and be represented by both a client object and a manager object. The reservation system 124 allows the manager to exchange communications, respond to requests for transactions, and conduct transactions with other managers.

[0046] The listing component 308 comprises program code for managers to list trip items, such as time-expiring inventory, for booking by clients. The listing component 308 is configured to receive the listing from a manager describing the inventory being offered; a timeframe of its availability including one or more of the start date, end date, start time, and an end time; a price; a geographical location; images and descriptions that characterize the inventory; and any other relevant information. For example, for an accommodation reservation system, a listing may include a type of accommodation (e.g., house, apartment, room, sleeping space, or other), a representation of its size (e.g., square footage, number of rooms), the dates that the accommodation is available, and a price (e.g., per night, per week, per month). The listing component 308 allows a user 106 to include additional information about the inventory, such as videos, photographs, and other media, or such as accessibility and other information.

[0047] The geographical location associated with the listing identifies the complete address, neighborhood, city, and/or country of the offered listing. The listing component 308 is also capable of converting one type of location

information (e.g., mailing address) into another type of location information (e.g., country, state, city, neighborhood) using externally available geographical map information.

[0048] The price of the listing is the amount of money a client needs to pay in order to complete a transaction for the inventory. The price may be specified as an amount of money per day, per week, per month, and/or per season, or per another interval of time specified by the manager. Additionally, the price may include additional charges such as cleaning fees, pet fees, service fees, and taxes, or the listing price may be listed separately from additional charges.

[0049] Each listing is represented in the reservation system 124 by a listing object, which includes the listing information as provided by the manager and a unique listing ID, both of which are stored in the listing store 318. Each listing object is also associated with the manager object for the manager providing the listing.

[0050] Each listing object has an associated listing calendar. The listing calendar stores the availability of the listing for each time interval in a period (each of which may be thought of as an independent item of time-expiring inventory), as specified by the manager or determined automatically (e.g., through a calendar import process). For example, a manager may access the listing calendar for a listing, and manually indicate the time intervals for which the listing is available for transaction by a client, which time intervals are blocked as not available by the manager, and which time intervals are already in transaction (e.g., booked) for a client. In addition, the listing calendar continues to store historical information as to the availability of the listing identifying which past time intervals were booked by clients, blocked, or available. Further, the listing calendar may include calendar rules (e.g., the minimum and maximum number of nights allowed for the inventory, a minimum or maximum number of nights needed between bookings, a minimum or maximum number of people allowed for the inventory). Information from each listing calendar is stored in the listing store 318.

[0051] The search component 310 comprises program code configured to receive an input search query from a client and return a set of time-expiring

inventory and/or listings that match the input query. Search queries are saved as query objects stored by the reservation system 124 in the query store 320. A query may contain a search location, a desired start time/date, a desired duration, a desired listing type, and a desired price range, and may also include other desired attributes or features of the listing. A potential client need not provide all the parameters of the query listed above in order to receive results from the search component 310. In some examples, the search component 310 provides a set of time-expiring inventory and/or listings in response to the submitted query to fulfill the parameters of the submitted query. The online system can also allow clients to browse listings without submitting a search query, in which case the viewing data recorded will only indicate that a client has viewed the particular listing without any further details from the submitted search query. Upon the client providing input selecting a time-expiring inventory/listing to more carefully review for possible transaction, the search component 310 records the selection/viewing data indicating which inventory/listing the client viewed. This information is also stored in the query store 320.

[0052] The search component 310 further comprises program code configured to receive an input request or search query from a client for one or more co-host to manage one or more listings in the online marketplace and return a set of ranked co-hosts relevant to the request, as described in further detail below.

[0053] The transaction component 312 comprises program code configured to enable clients to submit a contractual transaction request (also referred to as a formal request) to transact for time-expiring inventory. In operation, the transaction component 312 receives a transaction request from a client to transact for an item of time-expiring inventory, such as a particular date range for a listing offered by a particular manager. A transaction request may be a standardized request form that is sent by the client, which may be modified by responses to the request by the manager, either accepting or denying a received request form, such that agreeable terms are reached between the manager and the client. Modifications to a received request can include, for example, changing the date, price, or time/date range (and thus, effectively changing which time-expiring inventory is being transacted for). The standardized form may require

the client to record the start time/date, duration (or end time), or any other details that must be included for an acceptance to be binding without further communication.

[0054] The transaction component 312 receives the filled-out form from the client and, in one example, presents the completed request form including the booking parameters to the manager associated with the listing. The manager may accept the request, reject the request, or provide a proposed alternative that modifies one or more of the parameters. If the manager accepts the request (or the client accepts the proposed alternative), then the transaction component 312 updates an acceptance status associated with the request and the time-expiring inventory to indicate that the request was accepted. The client calendar and the listing calendar are also updated to reflect that the time-expiring inventory has been transacted on for a particular time interval. Other components not specifically described herein allow the client to complete payment and the manager to receive payment.

[0055] The transaction component 312 may further comprise code configured to enable clients to instantly book a listing, whereby the online marketplace books or reserves the listing upon receipt of the filled-out form from the client.

[0056] The transaction store 322 stores requests made by clients. Each request is represented by a request object. The request includes a timestamp, a requested start time, and a requested duration or reservation end time. Because the acceptance of a booking by a manager is a contractually binding agreement with the client that the manager will provide the time-expiring inventory to the client at the specified times, all the information that the manager needs to approve such an agreement is included in the request. A manager response to a request comprises a value indicating acceptance or denial and a timestamp. Other models may allow for instant booking, as mentioned above.

[0057] The transaction component 312 may also provide managers and clients with the ability to exchange informal requests to transact. Informal requests are not sufficient to be binding upon the client or manager if accepted, and, in terms of content, may vary from mere communications and general inquiries regarding the availability of inventory, to requests that fall just short of whatever specific

requirements the reservation system 124 sets forth for formal transaction requests. The transaction component 312 may also store informal requests in the transaction store 322, as both informal and formal requests provide useful information about the demand for time-expiring inventory.

[0058] The booking session store 324 stores booking session data for all booking sessions performed by clients. Booking session data may include details about a listing that was booked and data about one or more other listings that were viewed (or seriously considered) but not booked by the client before booking the listing. For example, once a listing is booked, the transaction component 312 may send data about the listing or the transaction, viewing data that was recorded for the booking session, and so forth, to be stored in the booking session store 324. The transaction component 312 may utilize other components or data stores to generate booking session data to be stored in the booking session store 324.

[0059] FIG. 4 is a flowchart illustrating aspects of a method 400 for a customized virtual agent system, according to some example embodiments. For illustrative purposes, the method 400 is described with respect to the networked system 100 of FIG. 1 and the block diagram of FIG. 2. It is to be understood that the method 400 may be practiced with other system configurations in other embodiments.

[0060] In operation 402, a computing system (e.g., server system 102 or customized virtual agent system 128) detects a cancelation of a reservation, by a host of a listing associated with the reservation, within a predefined time period from a start of the reservation. For example, the computing system can receive a notification for each reservation that is canceled by a host within a predefined time period, such as a day, 3 days, a week, from the start of a reservation. In another example, the computing system can continuously monitor a reservation system of the online marketplace to detect any cancelations and determine whether they are made by a host and within the predefined time period to trigger a rebooking virtual agent 208 (as explained above).

[0061] In operation 404, the computing system identifies, using a machine learning model trained to identify listings similar to a given listing in an online

marketplace, one or more alternate listings similar to the listing associated with the reservation that was canceled. For example, the computing system determines a reservation identifier, a listing associated with the reservation identifier and user (guest) information for the canceled reservation and inputs this data into the machine learning model to analyze and generate one or more alternate listings similar to the listing associated with the reservation identifier for the canceled reservation.

[0062] In some examples, the machine learning model identifies the one or more alternate listings similar to the listing associated with the reservation that was canceled by comparing details of the listing associated with the reservation (e.g., embeddings of the listing) that was canceled to each of a plurality of listings in the online marketplace (e.g., embeddings of each listing in the online marketplace) to determine a subset of the plurality of listings with a similarity score over a predefined threshold. In some examples, the one or more alternate listings are ranked according to a respective similarity score and provided to the computing device in ranked order. In some examples, only a predefined number (e.g., 3, 5) of the ranked alternative listings are provided to the computing device.

[0063] In operation 406, the computing system generates, using an LLM (e.g., rebooking virtual agent 208), details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and a calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled. In some examples, the LLM is trained on rebooking conversation history between customer support agents and users in the online marketplace, and customer support agent training materials for the online marketplace, to configure the large language model to provide a conversational tone consistent with the conversation history using terminology consistent within the online marketplace. For example, an LLM, such as DeepSeek, Alibaba, Llama, Mixtral, Flan-T5, and the like, that initially trained on massive datasets to learn general language understanding and patterns, this then fine-tuned by further training the pre-trained LLM on a focused dataset comprising the

rebooking conversation history between customer support agents and users in the online marketplace, and customer support agent training materials for the online marketplace. The generates a specialized LLM adapted to the online marketplace tone and terminology to adapt the pre-trained LLM to a specific task for handling a last-minute host cancelation scenario and to improve the performance for this task. The tone is particularly important in customer support scenarios which are typically managing various issues that have arisen during travel and terminology specific to the online marketplace is important so that terms are not mistranslated by a generic system. A dynamic prompt can then be generated to instruct the fine-tuned LLM to generate details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and a calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled in a language tone and using terminology specific to a given online marketplace.

[0064] Further, the LLM is configured to calculate a price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled. Since the reservation was canceled at the last minute, the price of alternate listings may be higher than the original listing of the canceled reservation. Accordingly, the LLM can calculate a custom price for each of the one or more listings that is discounted, if needed, to be in a same range of price as the listing of the canceled reservation. For example, the LLM can compare a potential reservation cost for each of the one or more alternate listings with the reservation cost of the canceled reservation. If any of the one or more alternate listings has a higher reservation cost (or over a predefined amount higher) than the canceled reservation cost, the LLM can calculate a custom price for those listings that discounts each of those listings so that each listing has a reservation price in the price range of the canceled reservation. The LLM can access coupon policy rules for the online marketplace to use to calculate the custom price.

[0065] In operation 408, the computing system provides to a computing device, the details for the cancelation of the reservation, details for the one or more

alternate listings similar to the reservation that was canceled, and calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled. In one example, the computing device is associated with a user that reserved the listing associated with the reservation that was canceled.

[0066] In some examples, this information is provided by the computing system to be displayed in a user interface on the computing device in a conversational manner that reflects the tone and terminology of the online marketplace. For example, the information can state "We have just been notified that your reservation for the apartment in San Francisco has been canceled by the host. To be sure that your travel plans are not interrupted, we have found the following accommodations in San Francisco that are available for the dates of your trip. We note that some of these accommodates typically have a higher price than the apartment that you originally booked and we have discounted the price to align with your original booking. Please select an alternate listing that works best for you or let me know if you would like to speak directly with a customer service agent. I am very sorry for any inconvenience that this change causes." It is to be understood that this is just one example of a conversation between the LLM (e.g., rebooking virtual agent 208) and a user to facilitate rebooking of a canceled reservation.

[0067] In another example, the computing device is associated with a customer support agent in the online marketplace. In this example, the LLM initiates a conversation with the customer support agent about the cancelation and the alternate listings for the customer support agent to review before the LLM directly contacts the user, or for the customer support agent to use to contact the user. The LLM can further initiate an outbound call with the customer support agent and user, if needed.

[0068] In operation 410, the computing device receives, from the computing device, a selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled. For example, a user or a customer support agent can select, via a user interface of the computing device, one of the alternate listings to book.

[0069] In operation 412, the computing system automatically reserves the first alternate listing and provides confirmation of the reservation for the first alternate listing to a user that reserved the listing associated with the reservation that was canceled. For example, the computing system causes the alternate listing to be reserved in the online marketplace and communicates confirmation of the reservation via a conversation with the user (e.g., guest and/or customer support agent), via email or text message, and/or other means.

[0070] In some examples, language translations services may be required in a call between a customer support agent and a user (e.g., guest and/or host). For example, a user may be traveling in Spain but only speak French and have a travel issue that needs resolution during the trip in Spain. As another example, a customer support agent may call a user about a last-minute cancelation and the user may not speak the same language as the customer support agent. FIG. 5 is a flowchart illustrating aspects of a method 500 for a customized virtual agent system, according to some example embodiments. For illustrative purposes, the method 500 is described with respect to the networked system 100 of FIG. 1 and the block diagram of FIG. 2. It is to be understood that the method 500 may be practiced with other system configurations in other embodiments.

[0071] In operation 502, a computing system (e.g., server system 102 or customized virtual agent system 128) receives audio input for a user in a call between two or more participants. For example, the computing system can initiate a call between a customer support agent of the online marketplace and a user that reserved the listing associated with the reservation that was canceled, in the rebooking scenario described above, before receiving the selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled or at any point during the rebooking process of FIG. 4. In another example, the computing system can detect in any already initiated or ongoing call, that translations services are needed. The computing system receives the audio input for a user in the call via the computing device used by the user for the call. In some examples, the computing initiates the translation virtual agent 206 to perform the operations shown in FIG. 5.

[0072] In operation 504, the computing system analyzes the audio input to detect

a language spoken by the user. For example, the computing system utilizes an LLM (e.g., the translation virtual agent 206), or other machine learning model or technique to detect the language spoken in the audio input.

[0073] In operation 506, the computing system determines that real-time translation is needed based on detecting that the language spoken by the user is not a language spoken by the customer support agent in the call. For example, the computing system compares the language spoken by the user to the language(s) spoken by the customer support agent, or by analyzing audio input by the customer support agent to determine the language spoken by the customer support agent.

[0074] In operation 508, the computing system, based on determining that real-time translation is needed, generates a translation of audio input received from the customer support agent in a detected language spoken by the user. In one example, the computing system initiates the translation virtual agent 206 configured as an LLM that uses a STT technique to analyze text of the audio input and generate a text translation of the audio input and then uses a TTS technique to generate audio output of the text translation. In another example, the LLM can translate the audio input directly into a translated audio output without translating it first to text.

[0075] In operation 510, the computing system customizes audio output of the translation based on detected language characteristics. The detected language characteristics can comprise at least one of an accent, a gender, or a regional dialect of the user. For example, the computing system can further analyze (e.g., via the LLM) the audio input to detect an accent, a gender, a regional dialect or other language or audio characteristic. For example, the LLM can be trained (or fine-tuned) to detect at least one of an accent or a regional dialect. For instance, the LLM can determine that the user is speaking Canadian French instead of European French and provide an audio translation of the audio input in Canadian French instead of European French. In this way the LLM customizes the audio output to match an accent, terminology, and other specific characteristics of the language that the user speaks. The computing system can further determine a gender of the speaker and provide an audio translation in a voice associated with

the speaker's gender. The computing system replaces audio input of the customer support agent with the customized audio output in at least one of the accent or gender of the user. In other examples, the computing system outputs the audio input of the customer support agent followed by the customized audio output.

[0076] Accordingly, the computing system, via the translation virtual agent 206, can provide real-time translation in a multi-person call on-demand as needed in a personalized voice based on the user's accent or regional dialect. Further, since the translation virtual agent 206 is within the customized virtual agent system 128 of the online marketplace, privacy will not be compromised.

[0077] FIG. 6 is a flowchart illustrating aspects of a method 600 for a customized virtual agent system, according to some example embodiments. For illustrative purposes, the method 600 is described with respect to the networked system 100 of FIG. 1 and the block diagram of FIG. 2. It is to be understood that the method 600 may be practiced with other system configurations in other embodiments.

[0078] In some examples, the computing system (e.g., server system 102 or customized virtual agent system 128) initiates an outbound call to a user. For example, the computing system can initiate an outbound call during the rebooking process described with respect to FIG. 4, such as before receiving the selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled or at another point in the process. In other examples, the computing system can initiate an outbound call based on a customer support issue identified as needing communication with a user. In some examples, the computing system initiates the voicemail detection virtual agent 210 to perform the operations of FIG. 6.

[0079] In operation 602, the computing system generates a prompt to instruct a LLM (e.g., the voicemail detection virtual agent 210) make an outbound call to a user. For example, a customer support issue may require a customer support agent to call a user. The computing system can detect the issue based on a message or audio input from the user or customer support agent, based on a customer support agent initiating a ticket for an issue, or other means. The

computing system generates a prompt that instructs an LLM to call the user and determine whether a live person or a voicemail system answers the outbound call.

[0080] In operation 604, the computing system detects, by the LLM, whether a live person or a voicemail system answers the outbound call. For example, the LLM analyzes the audio detected when placing the outbound call to determine if it is a live person or a voicemail system.

[0081] In operation 606, the computing device, based on detecting a live person answers the outbound call, returns the outbound call to a computing device of live customer support agent. In operation 608, the computing system based on detecting a voicemail system, provides a notification to the computing device of the live customer support agent that voicemail was reached. In this way, the computing system, via the LLM can place large numbers of outbound calls at any given moment in real-time from when each user issue is detected.

[0082] FIG. 7 is a flowchart depicting a machine learning pipeline 700, according to some examples. The machine learning pipeline 700 may be used to generate a trained model, such as the machine learning model configured to identify similar listings to a given listing, or the LLMs described herein, such as the translations virtual agent 206, the rebooking virtual agent 208 or the voicemail detection virtual agent 210.

[0083] Broadly, machine learning can involve using computer algorithms to automatically learn patterns and relationships in data, potentially without the need for explicit programming. Machine learning algorithms include three categories: supervised learning, unsupervised learning, and reinforcement learning.

- Supervised learning involves training a model using labeled data to predict an output for new, unseen inputs. Examples of supervised learning algorithms may include linear regression, decision trees, and neural networks.
- Unsupervised learning involves training a model on unlabeled data to find hidden patterns and relationships in the data. Examples of unsupervised

learning algorithms may include clustering, principal component analysis, and certain generative models, such as autoencoders.

- Reinforcement learning involves training a model to make decisions in a dynamic environment by receiving feedback in the form of rewards or penalties. Examples of reinforcement learning algorithms include Q-learning and policy gradient methods.

[0084] Examples of specific machine learning algorithms that may be deployed, according to some examples, include logistic regression, which is a type of supervised learning algorithm used for binary classification tasks. Logistic regression models the probability of a binary response variable based on one or more predictor variables. Another example type of machine learning algorithm is Naïve Bayes, which is a supervised learning algorithm used for classification tasks. Naïve Bayes is based on Bayes' theorem and assumes that the predictor variables are independent of each other. Random Forest is another type of supervised learning algorithm used for classification, regression, and other tasks. Random Forest builds a collection of decision trees and combines their outputs to make predictions.

[0085] Further examples include neural networks, which consist of interconnected layers of nodes (or neurons) that process information and make predictions based on the input data. Matrix factorization is another type of machine learning algorithm used for recommender systems and other tasks. Matrix factorization decomposes a matrix into two or more matrices to uncover hidden patterns or relationships in the data. Support Vector Machines (SVM) are a type of supervised learning algorithm used for classification, regression, and other tasks. SVM finds a hyperplane that separates the different classes in the data. Other machine learning algorithms may include decision trees, k-nearest neighbors, clustering algorithms, and deep learning algorithms, such as Convolutional Neural Networks (CNNs), recurrent neural networks (RNNs), and transformer models. The choice of algorithm may depend on various factors, such as the nature of the data, the complexity of the problem, and the performance requirements of the application. The performance of machine learning models may be evaluated on a separate test set of data that was not used

during training to ensure that the model can generalize to new, unseen data.

[0086] Two example types of problems in machine learning are classification problems and regression problems. Classification problems, also referred to as categorization problems, aim at classifying items into one of several category values (for example, is this object an apple or an orange?). Regression algorithms aim at quantifying some items (for example, by providing a value that is a real number).

[0087] Generating a trained machine learning model may include multiple phases that form part of the machine learning pipeline 700, including, for example, the following phases illustrated in FIG. 7. One phase includes data collection and preprocessing 702. This phase may include acquiring and cleaning data to ensure that it is suitable for use in the machine learning model. This phase may also include removing duplicates, handling missing values, and converting data into a suitable format.

[0088] For example, data is extracted from one or more databases 126 of an online marketplace that include data from a customer support system, as explained in further detail above.

[0089] In some examples, the data can be preprocessed using filters and downsampling to select a representative data set from the extracted data to be used for training data to train the machine learning model. Data can further be deduplicated to reduce the number of redundant information.

[0090] As another example, the training dataset can be downsampled to allow the machine learning model to learn more about positive instances.

[0091] The data resulting after any preprocessing is the training data used to train a model to generate the trained machine learning model.

[0092] Another phase illustrated in FIG. 7 is feature engineering 704. This phase can include selecting and transforming the training data to create features that are useful for predicting the target variable. Feature engineering may include (1) receiving features (e.g., as structured or labeled data in supervised learning) and/or (2) identifying features (e.g., unstructured or unlabeled data for unsupervised learning) in training data.

[0093] Another phase illustrated in FIG. 7 is model selection and training 706.

This phase can include selecting an appropriate machine learning algorithm and training it on the preprocessed data. This phase may further involve splitting the data into training and testing sets, using cross-validation to evaluate the model, and tuning hyperparameters to improve performance.

[0094] The model evaluation 708 phase can include evaluating the performance of a trained model on a separate testing dataset. This phase can help determine if the model is overfitting or underfitting and determine whether the model is suitable for deployment.

[0095] The prediction 710 phase involves using a trained to generate predictions on new, unseen data or to generate new data based on the training data. The validation, refinement or retraining 712 phase can include updating a model based on feedback generated from the prediction phase, such as new data or user feedback.

[0096] The deployment 714 phase can include integrating the trained model into a more extensive system or application, such as a web service, mobile app, or Internet of Things (IoT) device. This phase can involve setting up APIs, building a user interface, and ensuring that the model is scalable and can handle sufficiently large volumes of data. In some examples, adjustment can be performed prior to deployment 714.

[0097] FIG. 8 is a block diagram 800 illustrating a software architecture 802, which can be installed on any one or more of the devices described above. For example, in various embodiments, the client device 110 and server systems 130, 102, 120, 122, 124 and 128 may be implemented using some or all of the elements of the software architecture 802. FIG. 8 is merely a nonlimiting example of a software architecture, and it will be appreciated that many other architectures can be implemented to facilitate the functionality described herein. In various embodiments, the software architecture 802 is implemented by hardware such as a machine 900 of FIG. 9 that includes processors 910, memory 930, and input/output (I/O) components 950. In this example, the software architecture 802 can be conceptualized as a stack of layers where each layer may provide a particular functionality. For example, the software architecture 902 includes layers such as an operating system 804, libraries 806, frameworks 808,

and applications 810. Operationally, the applications 810 invoke API calls 812 through the software stack and receive messages 814 in response to the API calls 812, consistent with some embodiments.

[0098] In various implementations, the operating system 804 manages hardware resources and provides common services. The operating system 804 includes, for example, a kernel 820, services 822, and drivers 824. The kernel 820 acts as an abstraction layer between the hardware and the other software layers, consistent with some embodiments. For example, the kernel 820 provides memory management, processor management (e.g., scheduling), component management, networking, and security settings, among other functionalities. The services 822 can provide other common services for the other software layers. The drivers 824 are responsible for controlling or interfacing with the underlying hardware, according to some embodiments. For instance, the drivers 824 can include display drivers, camera drivers, BLUETOOTH® or BLUETOOTH® Low Energy drivers, flash memory drivers, serial communication drivers (e.g., Universal Serial Bus (USB) drivers), WI-FI® drivers, audio drivers, power management drivers, and so forth.

[0099] In some embodiments, the libraries 806 provide a low-level common infrastructure utilized by the applications 810. The libraries 806 can include system libraries 830 (e.g., C standard library) that can provide functions such as memory allocation functions, string manipulation functions, mathematic functions, and the like. In addition, the libraries 806 can include API libraries 832 such as media libraries (e.g., libraries to support presentation and manipulation of various media formats such as Moving Picture Experts Group-4 (MPEG4), Advanced Video Coding (H.264 or AVC), Moving Picture Experts Group Layer-3 (MP3), Advanced Audio Coding (AAC), Adaptive Multi-Rate (AMR) audio codec, Joint Photographic Experts Group (JPEG or JPG), or Portable Network Graphics (PNG)), graphics libraries (e.g., an OpenGL framework used to render graphic content in two dimensions (2D) and in three dimensions (3D) on a display), database libraries (e.g., SQLite to provide various relational database functions), web libraries (e.g., WebKit to provide web browsing functionality), and the like. The libraries 806 can also include a

wide variety of other libraries 834 to provide many other APIs to the applications 810.

[00100] The frameworks 808 provide a high-level common infrastructure that can be utilized by the applications 810, according to some embodiments. For example, the frameworks 808 provide various graphic user interface (GUI) functions, high-level resource management, high-level location services, and so forth. The frameworks 808 can provide a broad spectrum of other APIs that can be utilized by the applications 810, some of which may be specific to a particular operating system 804 or platform.

[00101] In an example embodiment, the applications 810 include a home application 850, a contacts application 852, a browser application 854, a book reader application 856, a location application 858, a media application 860, a messaging application 862, a game application 864, and a broad assortment of other applications, such as a third-party application 866. According to some embodiments, the applications 810 are programs that execute functions defined in the programs. Various programming languages can be employed to create one or more of the applications 810, structured in a variety of manners, such as object-oriented programming languages (e.g., Objective-C, Java, C++) or procedural programming languages (e.g., C or assembly language). In a specific example, the third-party application 866 (e.g., an application developed using the ANDROID™ or IOS™ software development kit (SDK) by an entity other than the vendor of the particular platform) may be mobile software running on a mobile operating system such as IOS™, ANDROID™, WINDOWS® Phone, or another mobile operating system. In this example, the third-party application 866 can invoke the API calls 812 provided by the operating system 804 to facilitate functionality described herein.

[00102] Some embodiments may particularly include a trip reservation application 867, which may be any application that requests data or other tasks to be performed by systems and servers described herein, such as the server system 102, third-party servers 130, and so forth. In certain embodiments, this may be a standalone application that operates to manage communications with a server system such as the third-party servers 130 or server system 102. In other

embodiments, this functionality may be integrated with another application. The trip reservation application 867 may request and display various data related to an online marketplace and may provide the capability for a user 106 to input data related to the system via voice, a touch interface, or a keyboard, or using a camera device of the machine 800, communication with a server system via the I/O components 850, and receipt and storage of object data in the memory 830. Presentation of information and user inputs associated with the information may be managed by the trip reservation application 867 using different frameworks 808, library 806 elements, or operating system 804 elements operating on a machine 800.

[00103] FIG. 9 is a block diagram illustrating components of a machine 900, according to some embodiments, able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, FIG. 9 shows a diagrammatic representation of the machine 900 in the example form of a computer system, within which instructions 916 (e.g., software, a program, an application 910, an applet, an app, or other executable code) for causing the machine 900 to perform any one or more of the methodologies discussed herein can be executed. In alternative embodiments, the machine 900 operates as a standalone device or can be coupled (e.g., networked) to other machines. In a networked deployment, the machine 900 may operate in the capacity of a server system 130, 102, 120, 122, 124, 128 and the like, or a client device 110 in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 900 can comprise, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions 916, sequentially or otherwise, that specify actions to be taken by the machine 900. Further, while only a single machine 900 is illustrated, the

term “machine” shall also be taken to include a collection of machines 900 that individually or jointly execute the instructions 916 to perform any one or more of the methodologies discussed herein.

[00104] In various embodiments, the machine 900 comprises processors 910, memory 930, and I/O components 950, which can be configured to communicate with each other via a bus 902. In an example embodiment, the processors 910 (e.g., a central processing unit (CPU), a reduced instruction set computing (RISC) processor, a complex instruction set computing (CISC) processor, a graphics processing unit (GPU), a digital signal processor (DSP), an application-specific integrated circuit (ASIC), a radio-frequency integrated circuit (RFIC), another processor, or any suitable combination thereof) include, for example, a processor 912 and a processor 914 that may execute the instructions 916. The term “processor” is intended to include multi-core processors 910 that may comprise two or more independent processors 912, 914 (also referred to as “cores”) that can execute instructions 916 contemporaneously. Although FIG. 9 shows multiple processors 910, the machine 900 may include a single processor 910 with a single core, a single processor 910 with multiple cores (e.g., a multi-core processor 910), multiple processors 912, 914 with a single core, multiple processors 912, 914 with multiple cores, or any combination thereof.

[00105] The memory 930 comprises a main memory 932, a static memory 934, and a storage unit 936 accessible to the processors 910 via the bus 902, according to some embodiments. The storage unit 936 can include a machine-readable medium 938 on which are stored the instructions 916 embodying any one or more of the methodologies or functions described herein. The instructions 916 can also reside, completely or at least partially, within the main memory 932, within the static memory 934, within at least one of the processors 910 (e.g., within the processor’s cache memory), or any suitable combination thereof, during execution thereof by the machine 900. Accordingly, in various embodiments, the main memory 932, the static memory 934, and the processors 910 are considered machine-readable media 938.

[00106] As used herein, the term “memory” refers to a machine-readable

medium 938 able to store data temporarily or permanently and may be taken to include, but not be limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, and cache memory. While the machine-readable medium 938 is shown, in an example embodiment, to be a single medium, the term "machine-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store the instructions 916. The term "machine-readable medium" shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions (e.g., instructions 916) for execution by a machine (e.g., machine 900), such that the instructions 916, when executed by one or more processors of the machine 900 (e.g., processors 910), cause the machine 900 to perform any one or more of the methodologies described herein. Accordingly, a "machine-readable medium" refers to a single storage apparatus or device, as well as "cloud-based" storage systems or storage networks that include multiple storage apparatus or devices. The term "machine-readable medium" shall accordingly be taken to include, but not be limited to, one or more data repositories in the form of a solid-state memory (e.g., flash memory), an optical medium, a magnetic medium, other nonvolatile memory (e.g., erasable programmable read-only memory (EPROM)), or any suitable combination thereof. The term "machine-readable medium" specifically excludes nonstatutory signals per se.

[00107] The I/O components 950 include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. In general, it will be appreciated that the I/O components 950 can include many other components that are not shown in FIG. 9. The I/O components 950 are grouped according to functionality merely for simplifying the following discussion, and the grouping is in no way limiting. In various example embodiments, the I/O components 950 include output components 952 and input components 954. The output components 952 include visual components (e.g., a display such as a plasma display panel (PDP), a light-emitting diode (LED) display, a liquid crystal display (LCD), a projector, or a cathode ray tube (CRT)), acoustic components

(e.g., speakers), haptic components (e.g., a vibratory motor), other signal generators, and so forth. The input components 954 include alphanumeric input components (e.g., a keyboard, a touch screen configured to receive alphanumeric input, a photo-optical keyboard, or other alphanumeric input components), point-based input components (e.g., a mouse, a touchpad, a trackball, a joystick, a motion sensor, or other pointing instruments), tactile input components (e.g., a physical button, a touch screen that provides location and force of touches or touch gestures, or other tactile input components), audio input components (e.g., a microphone), and the like.

[00108] In some further example embodiments, the I/O components 950 include biometric components 956, motion components 958, environmental components 960, or position components 962, among a wide array of other components. For example, the biometric components 956 include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, eye tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, electroencephalogram-based identification), and the like. The motion components 958 include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environmental components 960 include, for example, illumination sensor components (e.g., photometer), temperature sensor components (e.g., one or more thermometers that detect ambient temperature), humidity sensor components, pressure sensor components (e.g., barometer), acoustic sensor components (e.g., one or more microphones that detect background noise), proximity sensor components (e.g., infrared sensors that detect nearby objects), gas sensor components (e.g., machine olfaction detection sensors, gas detection sensors to detect concentrations of hazardous gases for safety or to measure pollutants in the atmosphere), or other components that may provide indications, measurements, or signals corresponding to a surrounding physical environment. The position components 962 include location sensor components (e.g., a GPS receiver component), altitude sensor components (e.g.,

altimeters or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

[00109] Communication can be implemented using a wide variety of technologies. The I/O components 950 may include communication components 964 operable to couple the machine 900 to a network 980 or devices 970 via a coupling 982 and a coupling 972, respectively. For example, the communication components 964 include a network interface component or another suitable device to interface with the network 980. In further examples, communication components 964 include wired communication components, wireless communication components, cellular communication components, near field communication (NFC) components, BLUETOOTH® components (e.g., BLUETOOTH® Low Energy), WI-FI® components, and other communication components to provide communication via other modalities. The devices 970 may be another machine 900 or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a USB).

[00110] Moreover, in some embodiments, the communication components 964 detect identifiers or include components operable to detect identifiers. For example, the communication components 964 include radio frequency identification (RFID) tag reader components, NFC smart tag detection components, optical reader components (e.g., an optical sensor to detect one-dimensional bar codes such as a Universal Product Code (UPC) bar code, multidimensional bar codes such as a Quick Response (QR) code, Aztec Code, Data Matrix, Dataglyph, MaxiCode, PDF417, Ultra Code, Uniform Commercial Code Reduced Space Symbology (UCC RSS)-2D bar codes, and other optical codes), acoustic detection components (e.g., microphones to identify tagged audio signals), or any suitable combination thereof. In addition, a variety of information can be derived via the communication components 964, such as location via Internet Protocol (IP) geo-location, location via WI-FI® signal triangulation, location via detecting a BLUETOOTH® or NFC beacon signal that may indicate a particular location, and so forth.

[00111] In various example embodiments, one or more portions of the network 980 can be an ad hoc network, an intranet, an extranet, a VPN, a LAN,

a WLAN, a WAN, a WWAN, a MAN, the internet, a portion of the internet, a portion of the PSTN, a plain old telephone service (POTS) network, a cellular telephone network, a wireless network, a WI-FI® network, another type of network, or a combination of two or more such networks. For example, the network 980 or a portion of the network 980 may include a wireless or cellular network, and the coupling 982 may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile (GSM) communications connection, or another type of cellular or wireless coupling. In this example, the coupling 982 can implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High-Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long-Term Evolution (LTE) standard, others defined by various standard-setting organizations, other long-range protocols, or other data transfer technology.

[00112] In example embodiments, the instructions 916 are transmitted or received over the network 980 using a transmission medium via a network interface device (e.g., a network interface component included in the communication components 964) and utilizing any one of a number of well-known transfer protocols (e.g., HTTP). Similarly, in other example embodiments, the instructions 916 are transmitted or received using a transmission medium via the coupling 972 (e.g., peer-to-peer coupling) to the devices 970. The term "transmission medium" shall be taken to include any intangible medium that is capable of storing, encoding, or carrying the instructions 916 for execution by the machine 900, and includes digital or analog communications signals or other intangible media to facilitate communication of such software.

[00113] Furthermore, the machine-readable medium 938 is nontransitory (in other words, not having any transitory signals) in that it does not embody a

propagating signal. However, labeling the machine-readable medium 938 "nontransitory" should not be construed to mean that the medium is incapable of movement; the machine-readable medium 938 should be considered as being transportable from one physical location to another. Additionally, since the machine-readable medium 938 is tangible, the machine-readable medium 938 may be considered to be a machine-readable device.

[00114] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[00115] Although an overview of the inventive subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader scope of embodiments of the present disclosure.

[00116] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[00117] As used herein, the term "or" may be construed in either an inclusive or exclusive sense. Moreover, plural instances may be provided for resources, operations, or structures described herein as a single instance.

Additionally, boundaries between various resources, operations, modules, components, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A computer-implemented method, comprising:

detecting a cancelation of a reservation, by a host of a listing associated with the reservation, within a predefined time period from a start of the reservation;

identifying, using a machine learning model trained to identify listings similar to a given listing in an online marketplace, one or more alternate listings similar to the listing associated with the reservation that was canceled;

generating, using a large language model, details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and a calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled, the large language model trained on rebooking conversation history between customer support agents and users in the online marketplace, and customer support agent training materials for the online marketplace, to configure the large language model to provide a conversational tone consistent with the conversation history and terminology consistent within the online marketplace;

providing, to a computing device, the details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled;

receiving, from the computing device, a selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled; and

automatically reserving the first alternate listing and providing confirmation of the reservation for the first alternate listing to a user that reserved the listing associated with the reservation that was canceled.

2. The computer-implemented method of claim 1, wherein identifying one or more alternate listings similar to the listing associated with the reservation that was canceled comprises comparing details of the listing associated with the reservation that was canceled to each of a plurality of listings in the online marketplace to determine a subset of the plurality of listings with a similarity score over a predefined threshold as the one or more alternate listings.

3. The computer-implemented method of claim 2, wherein the one or more alternate listings are ranked according to a respective similarity score and provided to the computing device in ranked order.

4. The computer-implemented method of claim 1, wherein the computing device is associated with a user that reserved the listing associated with the reservation that was canceled.

5. The computer-implemented method of claim 1, wherein the computing device is associated with a customer support agent in the online marketplace.

6. The computer-implemented method of claim 1, wherein before receiving the selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled, the method comprises:

causing a call to be initiated between a customer support agent of the online marketplace and a user that reserved the listing associated with the reservation that was canceled;

receiving audio input from the user in the call;

analyzing the audio input to detect a language spoken by the user;

determining that real-time translation is needed based on detecting that the language spoken by the user is not a language spoken by the customer support agent in the call;

based on determining that real-time translation is needed, generating a translation of audio input received from the customer support agent in a detected language spoken by the user; and

customizing audio output of the translation based on detected language characteristics, the detected language characteristics comprising at least one of an accent, a gender, or a regional dialect of the user.

7. The computer-implemented method of claim 6, wherein customizing the audio output comprises replacing audio input of the customer support agent with the customized audio output in at least one of the accent or gender of the user.

8. The computer-implemented method of claim 6, wherein customizing the audio output comprises outputting the audio input of the customer support agent followed by the customized audio output.

9. The computer-implemented method of claim 6, wherein the translation is generated using a second large language model trained to detect at least one of an accent of a user or a regional dialect of a user.

10. The computer-implemented method of claim 1, before receiving the selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled, the method comprises:

generating a prompt to instruct a second large language model make an outbound call to a user;

detecting, by the second large language model, whether a live person or a voicemail system answers the outbound call;

based on detecting a live person answers the outbound call, returning the outbound call to a computing device of live customer support agent; and

based on detecting a voicemail system, providing a notification to the computing device of the live customer support agent that voicemail was reached.

11. A computing system comprising:

a memory that stores instructions; and

one or more processors configured by the instructions to perform operations comprising:

detecting a cancelation of a reservation, by a host of a listing associated with the reservation, within a predefined time period from a start of the reservation;

identifying, using a machine learning model trained to identify listings similar to a given listing in an online marketplace, one or more alternate listings similar to the listing associated with the reservation that was canceled;

generating, using a large language model, details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and a calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled, the large language model trained on rebooking conversation history between customer support agents and users in the online marketplace, and customer support agent training materials for the online marketplace, to configure the large language model to provide a conversational tone consistent with the conversation history and terminology consistent within the online marketplace;

providing, to a computing device, the details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled;

receiving, from the computing device, a selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled; and

automatically reserving the first alternate listing and providing confirmation of the reservation for the first alternate listing to a user that reserved the listing associated with the reservation that was canceled.

12. The computing system of claim 11, wherein identifying one or more alternate listings similar to the listing associated with the reservation that was canceled comprises comparing details of the listing associated with the reservation that was canceled to each of a plurality of listings in the online marketplace to determine a subset of the plurality of listings with a similarity score over a predefined threshold as the one or more alternate listings.

13. The computing system of claim 12, wherein the one or more alternate listings are ranked according to a respective similarity score and provided to the computing device in ranked order.

14. The computing system of claim 11, wherein the computing device is associated with a user that reserved the listing associated with the reservation that was canceled.

15. The computing system of claim 11, wherein the computing device is associated with a customer support agent in the online marketplace.

16. The computing system of claim 11, wherein before receiving the selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled, the operations comprise:

causing a call to be initiated between a customer support agent of the online marketplace and a user that reserved the listing associated with the reservation that was canceled;

receiving audio input from the user in the call;

analyzing the audio input to detect a language spoken by the user;

determining that real-time translation is needed based on detecting that the language spoken by the user is not a language spoken by the customer support agent in the call;

based on determining that real-time translation is needed, generating a translation of audio input received from the customer support agent in a detected language spoken by the user; and

customizing audio output of the translation based on detected language characteristics, the detected language characteristics comprising at least one of an accent, a gender, or a regional dialect of the user.

17. The computing system of claim 16, wherein customizing the audio output comprises replacing audio input of the customer support agent with the customized audio output in at least one of the accent or gender of the user.

18. The computing system of claim 16, wherein customizing the audio output comprises outputting the audio input of the customer support agent followed by the customized audio output or wherein the translation is generated using a second large language model trained to detect at least one of an accent of a user or a regional dialect of a user.

19. The computing system of claim 11, before receiving the selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled, the operations comprise:

generating a prompt to instruct a second large language model make an outbound call to a user;

detecting, by the second large language model, whether a live person or a voicemail system answers the outbound call;

based on detecting a live person answers the outbound call, returning the outbound call to a computing device of live customer support agent; and

based on detecting a voicemail system, providing a notification to the computing device of the live customer support agent that voicemail was reached.

20. A non-transitory computer-readable medium comprising instructions stored thereon that are executable by at least one processor to cause a computing device to perform operations comprising:

detecting a cancelation of a reservation, by a host of a listing associated with the reservation, within a predefined time period from a start of the reservation;

identifying, using a machine learning model trained to identify listings similar to a given listing in an online marketplace, one or more alternate listings similar to the listing associated with the reservation that was canceled;

generating, using a large language model, details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and a calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled, the large language model trained on rebooking conversation history between customer support agents and users in the online marketplace, and customer support agent training materials for the online marketplace, to configure the large language model to provide a conversational tone consistent with the conversation history and terminology consistent within the online marketplace;

providing, to a computing device, the details for the cancelation of the reservation, details for the one or more alternate listings similar to the reservation that was canceled, and calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled;

receiving, from the computing device, a selection of a first alternate listing of the one or more alternate listings similar to the reservation that was canceled; and

automatically reserving the first alternate listing and providing confirmation of the reservation for the first alternate listing to a user that reserved the listing associated with the reservation that was canceled.

ABSTRACT

Systems and methods are provided for generating, using a large language model, details for a cancelation of the reservation, details for the one or more alternate listings, and a calculated price for each of the one or more alternate listings based on discount parameters and details of the listing associated with the reservation that was canceled. In some examples, the large language model is trained on rebooking conversation history between customer support agents and users in the online marketplace, and customer support agent training materials for the online marketplace, to configure the large language model to provide a conversational tone consistent with the conversation history and terminology consistent within the online marketplace. The systems and methods provide these details to a computing device, receive a selection of a first alternate listing of the and automatically reserve the first alternate listing.