

(51) International Patent Classification:

G06F 40/35 (2020.01) H04L 51/216 (2022.01)

H04L 51/02 (2022.01)

(21) International Application Number:

PCT/US2024/032878

(22) International Filing Date:

06 June 2024 (06.06.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/472,290 10 June 2023 (10.06.2023) US

18/677,723 29 May 2024 (29.05.2024) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:

US 18/677,723 (CON)

Filed on 29 May 2024 (29.05.2024)

(71) Applicant: SNAP INC. [US/US]; 3000 31st Street, Santa Monica, California 90405 (US).

(72) Inventors: SUN, Haowen; 3000 31st Street, Santa Monica, California 90405 (US). MULLIGAN, William Spencer; 3000 31st Street, Santa Monica, California 90405 (US). BOYD, Nathan Kenneth; 3000 31st Street, Santa Monica, California 90405 (US). KIM, Hee Hun; 3000 31st Street, Santa Monica, California 90405 (US). ISHCENKO, Dmytro; 3000 31st Street, Santa Monica, California 90405 (US). HINKELDEY, Lily; 3000 31st Street, Santa Monica, California 90405 (US). BREWER, Jason; 3000 31st Street, Santa Monica, California 90405 (US). MELBYE, Charles; 3000 31st Street, Santa Monica, California 90405

(54) Title: CHATBOT LONG-TERM MEMORY

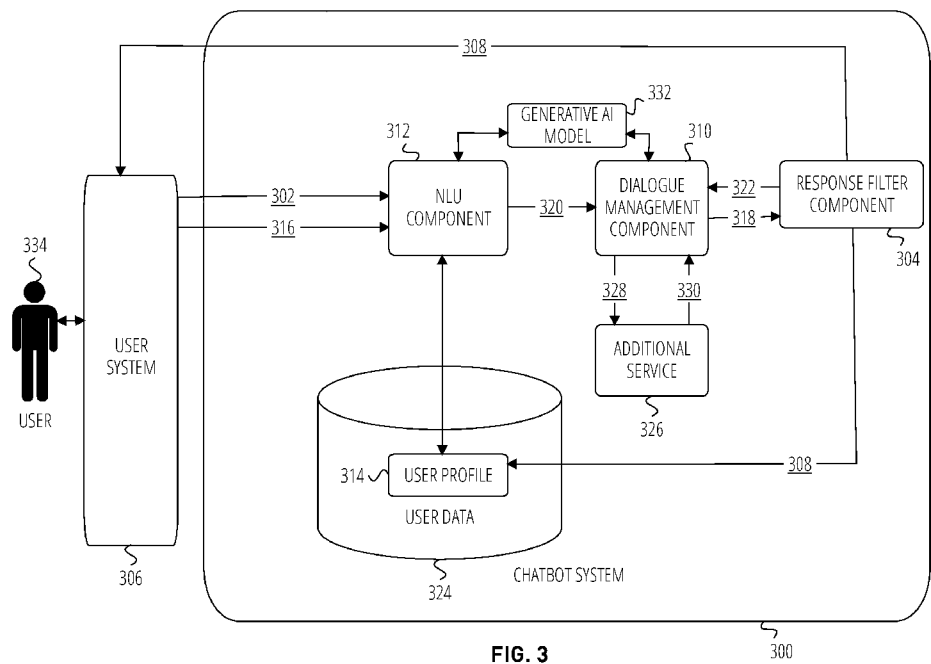


FIG. 3

(57) Abstract: A chatbot system for an interactive platform is disclosed. The chatbot system retrieves a conversation history of one or more conversations between a user and a chatbot from a conversation history datastore and generates one or more summarized memories using the conversation history. One or more moderated memories are generated using the summarized memories. The moderated memories are stored in a memories datastore. A user prompt is received, and a current conversation context is generated from a current conversation between the user and the chatbot. One or more memories are retrieved from the memories datastore using the current conversation context. An augmented prompt is generated using the user prompt and the one or more memories, which is communicated to a generative AI model. A response is received from the generative AI model to the augmented prompt, which is provided to the user.

(US). **MASHRABOV, Aleksandr**; 3000 31st Street, Santa Monica, California 90405 (US).

(74) **Agent: PERDOK, Monique M.** et al.; P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

(81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

CHATBOT LONG-TERM MEMORY

CLAIM OF PRIORITY

[0001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 63/472,290, filed on June 10, 2023, and U.S. Patent Application Serial No. 18/677,723, filed on May 29, 2024, each of which are incorporated herein by reference in their entireties.

TECHNICAL FIELD

[0002] The present disclosure relates generally to interactive platforms and more particularly to providing interactive interfaces to users of an interactive platform.

BACKGROUND

[0003] Users enjoy accessing interactive platforms to share content with other users of the interactive platform. In addition, users often provide sensitive content when communicating with a chatbot.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0004] In the drawings, which are not necessarily drawn to scale, like numerals may describe similar components in different views. To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced. Some non-limiting examples are illustrated in the figures of the accompanying drawings in which:

[0005] FIG. 1 is a diagrammatic representation of a networked environment in which the present disclosure may be deployed, according to some examples.

[0006] FIG. 2 is a diagrammatic representation of a machine in the form of a computer system within which a set of instructions may be executed to cause the machine to perform any one or more of the methodologies discussed herein, according to some examples.

[0007] FIG. 3 is a block diagram of a chatbot system in accordance with some examples.

[0008] FIG. 4A is a process flow diagram of a generative AI model prompting method, in accordance with some examples.

[0009] FIG. 4B is a collaboration diagram of the operations of a chatbot system, in accordance with some examples,

[0010] FIG. 5 is an illustration of a chat session between a user and a chatbot, in accordance with some examples.

[0011] FIG. 6A is a collaboration diagram of components of a chatbot system, in accordance with some examples.

[0012] FIG. 6B is an activity diagram of a conversation history method of a chatbot system, in accordance with some examples.

[0013] FIG. 7 illustrates a machine-learning pipeline, according to some examples.

[0014] FIG. 8 illustrates training and use of a machine-learning program, according to some examples.

[0015] FIG. 9 is a diagrammatic representation of a data structure as maintained in a database, according to some examples.

[0016] FIG. 10 is a diagrammatic representation of a message, according to some examples.

[0017] FIG. 11 is a flowchart for an access-limiting process, according to some examples.

[0018] FIG. 12 is a diagrammatic representation of a messaging system, according to some examples, that has both client-side and server-side functionality.

[0019] FIG. 13 is a block diagram showing a software architecture within which examples may be implemented.

DETAILED DESCRIPTION

[0020] Interactive platforms (e.g., interaction systems, social platforms, social media platforms, AR platforms, applications, messaging platforms, AR applications, operating systems, gaming systems or applications, systems with which a user interacts, and the like) may provide a way for users to interact with other users. For some users, interaction with an interaction system may be enhanced by having a chatbot to interact with. The chatbot can serve multiple uses, including providing a way for a user obtain information. The more accurate the information is, the more useful that information is to a user.

[0021] In some chatbots, a generative Artificial Intelligence (AI) model such as, but not limited to, a Large Language Model (LLM) or the like, used to perform Natural Language Generation (NLG) also performs personalization and memorizing of the generative AI model's responses to user prompts. This effect is achieved via the generative AI model's ability to remember users' personal information from previous conversations. However, outcomes can be unpredictable. For example, if users ask for movie recommendations right after telling a chatbot they like to watch comedy, the chatbot can recommend comedy movies. However, if there're some other irrelevant conversations in between, the generative AI model will quickly forget users' preference for comedy. Besides, upon every new session, knowledge from conversations in the last session will be completely forgotten. That is, there's no information persistence across multi-session chats. Solely relying on a generative AI model for memorizing, a provider of a chatbot lacks control over how long to memorize, and what to memorize. In addition, memorizing is not always desirable. Users might mention embarrassing incidents to a chatbot with no expectation that they're being remembered and referred to in the future as the chatbot provider may not control how the memorized information is used in current conversations.

[0022] In some examples, a chatbot system enables personalized responses. By retrieving and utilizing a user's conversation history and memories, the chatbot can generate more personalized and contextually relevant responses. This creates a more natural conversation flow.

[0023] In some examples, a chatbot system provides a persistent memory. The chatbot system summarizes and stores memories from past conversations in a persistent datastore. This allows the chatbot system to "remember" user facts, interests, and preferences over multiple sessions.

[0024] In some examples, a chatbot system features a controlled memory. The chatbot system moderates memories before storing memories to filter out unwanted content. A provider of an interactive platform hosting the chatbot system can control what is remembered. This prevents unpredictable responses.

[0025] In some examples, a chatbot system uses augmented prompting of a generative AI model. Using stored user memories, the chatbot system augments

prompts to a generative AI model. This allows the model to leverage long-term user memory to improve responses.

[0026] In some examples, a chatbot system enables isolated memories.

Memories are isolated per user or conversation group. One group's memories won't affect another's responses. This increases privacy for a user.

[0027] In some examples, a chatbot system provides for a deletable conversation history. Users can request deletion of their conversation history and associated memories. This gives users control over their data.

[0028] In some examples, a chatbot system provides improved scalability. Storing summarized memories allows the chatbot system to scale to long conversations while reducing compute costs compared to relying solely on a generative model.

[0029] In some examples, the chatbot system receives a user prompt from the user, determines a conversation history of one or more conversations between a user and a chatbot, generates an augmented prompt using the user prompt and the conversation history, communicates the augmented prompt to a generative AI model, receives a response from the generative AI model to the augmented prompt, and provides the response to the user.

[0030] In some examples, the chatbot system determines the conversation history by retrieving, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot.

[0031] In some examples, the chatbot system generates one or more summarized memories using the conversation history.

[0032] In some examples, the chatbot system generates one or more moderated memories using the summarized memories and stores the one or more moderated memories into a memories datastore.

[0033] In some examples, the chatbot system receives a request from the user to delete the conversation history and in response to receiving the request, deletes the conversation history and one or more memories stored in the memories datastore associated with the conversation history.

[0034] In some examples, the chatbot system generates the augmented prompt by generating a current conversation context from a current conversation between the user and the chatbot, retrieving, from the memories datastore, one or more

stored memories using the current conversation context, and generating the augmented prompt using the one or more stored memories.

[0035] In some examples, the one or more memories stored in the memories datastore are isolated for a first conversation group, and the one or more memories are not used by the chatbot system to generate the augmented prompt for a second conversation group.

[0036] In some examples, the methodologies described herein relate to a method of an interactive platform where the method includes: retrieving, by one or more processors, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot; generating one or more summarized memories using the chat history; generating one or more moderated memories using the summarized memories; storing the one or more moderated memories into a memories datastore storing one or more memories; receiving a user prompt from the user; generating a current conversation context from a current conversation between the user and the chatbot; retrieving, by one or more processors, from the memories datastore, one or more memories using the current conversation context; generating an augmented prompt using the user prompt and the one or more memories; communicating the augmented prompt to a generative AI model; receiving a response from the generative AI model to the augmented prompt; and providing the response to the user.

[0037] In some examples, the methodologies described herein relate to a machine where the machine includes: one or more processors; and a memory storing instructions that, when executed by the one or more processors, cause the machine to perform operations including: retrieving from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot; generating one or more summarized memories using the chat history; generating one or more moderated memories using the summarized memories; storing the one or more moderated memories into a memories datastore storing one or more memories; receiving a user prompt from the user; generating a current conversation context from a current conversation between the user and the chatbot; retrieving, by one or more processors, from the memories datastore, one or more memories using the current conversation context; generating an augmented prompt using the user prompt and the one or more memories;

communicating the augmented prompt to a generative AI model; receiving a response from the generative AI model to the augmented prompt; and providing the response to the user.

[0038] In some examples, the methodologies described herein relate to a machine-readable medium storing machine-executable instructions that, when executed by a machine, cause the machine to perform operations including: retrieving, by one or more processors, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot; generating one or more summarized memories using the chat history; generating one or more moderated memories using the summarized memories; storing the one or more moderated memories into a memories datastore storing one or more memories; receiving a user prompt from the user; generating a current conversation context from a current conversation between the user and the chatbot; retrieving, by one or more processors, from the memories datastore, one or more memories using the current conversation context; generating an augmented prompt using the user prompt and the one or more memories; communicating the augmented prompt to a generative AI model; receiving a response from the generative AI model to the augmented prompt; and providing the response to the user.

[0039] Other technical features may be readily apparent to one skilled in the art from the following figures, and descriptions.

NETWORKED COMPUTING ENVIRONMENT

[0040] FIG. 1 is a block diagram showing an example interactive platform 100 including one or more interaction components for facilitating interactions (e.g., exchanging text messages, conducting text audio and video calls, or playing games) over a network. The interactive platform 100 includes multiple user systems 102, each of which hosts multiple applications, including an interaction client 104 and other applications 106. Each interaction client 104 is communicatively coupled, via one or more communication networks including a network 108 (e.g., the Internet), to other instances of the interaction client 104 (e.g., hosted on respective other user systems 102), an interactive server system 110 and third-party servers 112). An interaction client 104 can also communicate

with locally hosted applications 106 using Applications Program Interfaces (APIs).

[0041] Each user system 102 may include multiple user devices, such as a mobile device 114, head-wearable apparatus 116, and a computer user device 118 that are communicatively connected to exchange data and messages.

[0042] An interaction client 104 interacts with other interaction clients 104 and with the interactive server system 110 via the network 108. The data exchanged between the interaction clients 104 (e.g., interactions 120) and between the interaction clients 104 and the interactive server system 110 includes functions (e.g., commands to invoke functions) and payload data (e.g., text, audio, video, or other multimedia data).

[0043] The interactive server system 110 provides server-side functionality via the network 108 to the interaction clients 104. While certain functions of the interactive platform 100 are described herein as being performed by either an interaction client 104 or by the interactive server system 110, the location of certain functionality either within the interaction client 104 or the interactive server system 110 may be a design choice. For example, it may be technically preferable to initially deploy particular technology and functionality within the interactive server system 110 but to later migrate this technology and functionality to the interaction client 104 where a user system 102 has sufficient processing capacity.

[0044] The interactive server system 110 supports various services and operations that are provided to the interaction clients 104. Such operations include transmitting data to, receiving data from, and processing data generated by the interaction clients 104. This data may include message content, user device information, geolocation information, media augmentation and overlays, message content persistence conditions, interactive platform information, and live event information. Data exchanges within the interactive platform 100 are invoked and controlled through functions available via user interfaces (UIs) of the interaction clients 104.

[0045] Turning now specifically to the interactive server system 110, an API server 122 is coupled to and provides programmatic interfaces to Interactive servers 124, making the functions of the Interactive servers 124 accessible to

interaction clients 104, other applications 106 and third-party server 112. The Interactive servers 124 are communicatively coupled to a database server 126, facilitating access to a database 128 that stores data associated with interactions processed by the Interactive servers 124. Similarly, a web server 130 is coupled to the Interactive servers 124 and provides web-based interfaces to the Interactive servers 124. To this end, the web server 130 processes incoming network requests over the Hypertext Transfer Protocol (HTTP) and several other related protocols.

[0046] The API server 122 receives and transmits interaction data (e.g., commands and message payloads) between the Interactive servers 124 and the user systems 102 (and, for example, interaction clients 104 and other application 106) and the third-party server 112. Specifically, the API server 122 provides a set of interfaces (e.g., routines and protocols) that can be called or queried by the interaction client 104 and other applications 106 to invoke functionality of the Interactive servers 124. The API server 122 exposes various functions supported by the Interactive servers 124, including account registration; login functionality; the sending of interaction data, via the Interactive servers 124, from a particular interaction client 104 to another interaction client 104; the communication of media files (e.g., images or video) from an interaction client 104 to the Interactive servers 124; the settings of a collection of media data (e.g., a story); the retrieval of a list of friends of a user of a user system 102; the retrieval of messages and content; the addition and deletion of entities (e.g., friends) to an entity graph (e.g., a social graph); the location of friends within a social graph; and opening an application event (e.g., relating to the interaction client 104).

[0047] The Interactive servers 124 host multiple systems and subsystems, described below with reference to FIG. 12.

LINKED APPLICATIONS

[0048] Returning to the interaction client 104, features and functions of an external resource (e.g., a linked application 106 or applet) are made available to a user via an interface of the interaction client 104. In this context, “external” refers to the fact that the application 106 or applet is external to the interaction client 104. The external resource is often provided by a third party but may also be provided by the creator or provider of the interaction client 104. The interaction client 104 receives a user selection of an option to launch or access features of

such an external resource. The external resource may be the application 106 installed on the user system 102 (e.g., a “native app”), or a small-scale version of the application (e.g., an “applet”) that is hosted on the user system 102 or remote of the user system 102 (e.g., on third-party servers 112). The small-scale version of the application includes a subset of features and functions of the application (e.g., the full-scale, native version of the application) and is implemented using a markup-language document. In some examples, the small-scale version of the application (e.g., an “applet”) is a web-based, markup-language version of the application and is embedded in the interaction client 104. In addition to using markup-language documents (e.g., a *.ml file), an applet may incorporate a scripting language (e.g., a *.js file or a .json file) and a style sheet (e.g., a *.ss file).

[0049] In response to receiving a user selection of the option to launch or access features of the external resource, the interaction client 104 determines whether the selected external resource is a web-based external resource or a locally-installed application 106. In some cases, applications 106 that are locally installed on the user system 102 can be launched independently of and separately from the interaction client 104, such as by selecting an icon corresponding to the application 106 on a home screen of the user system 102. Small-scale versions of such applications can be launched or accessed via the interaction client 104 and, in some examples, no or limited portions of the small-scale application can be accessed outside of the interaction client 104. The small-scale application can be launched by the interaction client 104 receiving, from a third-party server 112 for example, a markup-language document associated with the small-scale application and processing such a document.

[0050] In response to determining that the external resource is a locally-installed application 106, the interaction client 104 instructs the user system 102 to launch the external resource by executing locally-stored code corresponding to the external resource. In response to determining that the external resource is a web-based resource, the interaction client 104 communicates with the third-party servers 112 (for example) to obtain a markup-language document corresponding to the selected external resource. The interaction client 104 then processes the obtained markup-language document to present the web-based external resource within a user interface of the interaction client 104.

[0051] The interaction client 104 can notify a user of the user system 102, or other users related to such a user (e.g., “friends”), of activity taking place in one or more external resources. For example, the interaction client 104 can provide participants in a conversation (e.g., a chat session) in the interaction client 104 with notifications relating to the current or recent use of an external resource by one or more members of a group of users. One or more users can be invited to join in an active external resource or to launch a recently-used but currently inactive (in the group of friends) external resource. The external resource can provide participants in a conversation, each using respective interaction clients 104, with the ability to share an item, status, state, or location in an external resource in a chat session with one or more members of a group of users. The shared item may be an interactive chat card with which members of the chat can interact, for example, to launch the corresponding external resource, view specific information within the external resource, or take the member of the chat to a specific location or state within the external resource. Within a given external resource, response messages can be sent to users on the interaction client 104. The external resource can selectively include different media items in the responses, based on a current context of the external resource.

[0052] The interaction client 104 can present a list of the available external resources (e.g., applications 106 or applets) to a user to launch or access a given external resource. This list can be presented in a context-sensitive menu. For example, the icons representing different ones of the application 106 (or applets) can vary based on how the menu is launched by the user (e.g., from a conversation interface or from a non-conversation interface).

MACHINE ARCHITECTURE

[0053] FIG. 2 is a diagrammatic representation of the machine 200 within which instructions 202 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 200 to perform any one or more of the methodologies discussed herein may be executed. For example, the instructions 202 may cause the machine 200 to execute any one or more of the methods described herein. The instructions 202 transform the general, non-programmed machine 200 into a particular machine 200 programmed to carry out the described and illustrated functions in the manner described. The machine

200 may operate as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 200 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 200 may include, but not be limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smartphone, a mobile device, a wearable device (e.g., a smartwatch), 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 202, sequentially or otherwise, that specify actions to be taken by the machine 200. Further, while a single machine 200 is illustrated, the term “machine” shall also be taken to include a collection of machines that individually or jointly execute the instructions 202 to perform any one or more of the methodologies discussed herein. The machine 200, for example, may include the user system 102 or any one of multiple server devices forming part of the interactive server system 110. In some examples, the machine 200 may also include both client and server systems, with certain operations of a particular method or algorithm being performed on the server-side and with certain operations of the particular method or algorithm being performed on the client-side.

[0054] The machine 200 may include processors 204, memory 206, and input/output I/O components 208, which may be configured to communicate with each other via a bus 210. In an example, the processors 204 (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) may include, for example, a processor 212 and a processor 214 that execute the instructions 202. The term “processor” is intended to include multi-core processors that may include two or more independent processors (sometimes referred to as “cores”) that may execute instructions contemporaneously. Although FIG. 2 shows multiple processors 204,

the machine 200 may include a single processor with a single-core, a single processor with multiple cores (e.g., a multi-core processor), multiple processors with a single core, multiple processors with multiples cores, or any combination thereof.

[0055] The memory 206 includes a main memory 216, a static memory 218, and a storage unit 220, both accessible to the processors 204 via the bus 210. The main memory 216, the static memory 218, and storage unit 220 store the instructions 202 embodying any one or more of the methodologies or functions described herein. The instructions 202 may also reside, completely or partially, within the main memory 216, within the static memory 218, within machine-readable medium 222 within the storage unit 220, within at least one of the processors 204 (e.g., within the processor's cache memory), or any suitable combination thereof, during execution thereof by the machine 200.

[0056] The I/O components 208 may include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. The specific I/O components 208 that are included in a particular machine will depend on the type of machine. For example, portable machines such as mobile phones may include a touch input device or other such input mechanisms, while a headless server machine will likely not include such a touch input device. It will be appreciated that the I/O components 208 may include many other components that are not shown in FIG.

2. In various examples, the I/O components 208 may include user output components 224 and user input components 226. The user output components 224 may 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, resistance mechanisms), other signal generators, and so forth. The user input components 226 may 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 another pointing instrument), 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.

[0057] In further examples, the I/O components 208 may include biometric components 228, motion components 230, environmental components 232, or position components 234, among a wide array of other components. For example, the biometric components 228 include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, or eye-tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, or brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, or electroencephalogram-based identification), and the like. The motion components 230 include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope). The biometric components may include a brain-machine interface (BMI) system that allows communication between the brain and an external device or machine. This is achieved by recording brain activity, translating it into a format that can be understood by a computer, and then using the resulting signals to control the device or machine.

[0058] Example types of BMI technologies, include, but are not limited to:

- Electroencephalography (EEG) based BMIs, which record electrical activity in the brain using electrodes placed on the scalp.
- Invasive BMIs, which involve surgically implanting electrodes directly into the brain.
- Optogenetics BMIs, which use light to control the activity of specific nerve cells in the brain.
- Functional magnetic resonance imaging (fMRI)-based BMIs, which use magnetic fields to measure blood flow in the brain, which can be used to infer brain activity.

[0059] Any BMI data collected by an interactive platform is captured and stored with only user approval and deleted on user request. Further, such data may be used for very limited purposes, such as identity verification or user input. To

ensure limited and authorized use of BMI data and other personally identifiable information (PII), access to this BMI data is restricted to authorized personnel only, if at all. Any use of BMI data may strictly be limited to a designated purpose, and the BMI data is not shared or sold to any third party without the explicit consent of the user. In addition, appropriate technical and organizational measures are implemented to ensure the security and confidentiality of this sensitive information.

[0060] The environmental components 232 include, for example, one or more cameras (with still image/photograph and video capabilities), 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 sensors (e.g., gas detection sensors to detection 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.

[0061] With respect to cameras, the user system 102 may have a camera system including, for example, front cameras on a front surface of the user system 102 and rear cameras on a rear surface of the user system 102. The front cameras may, for example, be used to capture still images and video of a user of the user system 102 (e.g., “selfies”), which may then be augmented with augmentation data (e.g., filters) described above. The rear cameras may, for example, be used to capture still images and videos in a more traditional camera mode, with these images similarly being augmented with augmentation data. In addition to front and rear cameras, the user system 102 may also include a 360° camera for capturing 360° photographs and videos.

[0062] Further, the camera system of the user system 102 may include dual rear cameras (e.g., a primary camera as well as a depth-sensing camera), or even triple, quad or penta rear camera configurations on the front and rear sides of the user system 102. These multiple cameras systems may include a wide camera, an

ultra-wide camera, a telephoto camera, a macro camera, and a depth sensor, for example.

[0063] The position components 234 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.

[0064] Communication may be implemented using a wide variety of technologies. The I/O components 208 further include communication components 236 operable to couple the machine 200 to a network 238 or devices 240 via respective coupling or connections. For example, the communication components 236 may include a network interface component or another suitable device to interface with the network 238. In further examples, the communication components 236 may 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 240 may be another machine or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a USB).

[0065] Moreover, the communication components 236 may detect identifiers or include components operable to detect identifiers. For example, the communication components 236 may 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 Universal Product Code (UPC) bar code, multi-dimensional bar codes such as Quick Response (QR) code, Aztec code, Data Matrix, Dataglyph, MaxiCode, PDF417, Ultra Code, UCC RSS-2D bar code, and other optical codes), or acoustic detection components (e.g., microphones to identify tagged audio signals). In addition, a variety of information may be derived via the communication components 236, such as location via Internet Protocol (IP) geolocation, location via Wi-Fi® signal triangulation, location via detecting an NFC beacon signal that may indicate a particular location, and so forth.

[0066] The various memories (e.g., main memory 216, static memory 218, and memory of the processors 204) and storage unit 220 may store one or more sets of instructions and data structures (e.g., software) embodying or used by any one or more of the methodologies or functions described herein. These instructions (e.g., the instructions 202), when executed by processors 204, cause various operations to implement the disclosed examples.

[0067] The instructions 202 may be transmitted or received over the network 238, using a transmission medium, via a network interface device (e.g., a network interface component included in the communication components 236) and using any one of several well-known transfer protocols (e.g., hypertext transfer protocol (HTTP)). Similarly, the instructions 202 may be transmitted or received using a transmission medium via a coupling (e.g., a peer-to-peer coupling) to the devices 240.

[0068] FIG. 3 is a block diagram of a chatbot system 300, in accordance with some examples. A chatbot system 300 is conducts a chat session with a user 334 during an interactive session.

[0069] In some examples, the chatbot system 300 is a software platform designed to simulate human conversation through voice commands or text chats. The chatbot system 300 may employ natural language processing (NLP) and machine learning (ML)/artificial intelligence methodologies to understand and interpret the input of the user 334 and generate a chatbot response 308.

[0070] In some examples, the chatbot's architecture includes one or more natural language understanding (NLU) components, such as NLU component 312, and one or more dialogue management components, such as dialogue management component 310. The NLU component 312 is responsible for understanding an intent 320 of the user 334 and extracting relevant information from the input of the user, such as prompt 302. This is achieved by analyzing the prompt 302 and mapping it to an intent. The NLU component 312 can use various methodologies such as rule-based systems, statistical models, LLMs, neural networks, and the like to understand the input of the user. The dialogue management component 310 generates a chatbot response 308 to the input of the user. The dialogue management component 310 uses the intent 320 and any extracted information from the NLU component 312 to determine the appropriate chatbot response 308.

This can be done using rule-based systems, decision trees, statistical models, LLMs, neural networks, and the like. In some examples, the NLU component 312 and the dialogue management component 310 are a single component.

[0071] Once the NLU component 312 extracts the relevant information from the prompt 302 as an intent 320 of the user 334, the dialogue management component 310 receives an appropriate chatbot response 308. The dialogue management component 310 generates responses in a human-like manner by using methodologies such as, but not limited to, text generation and machine learning models.

[0072] In some examples, either the NLU component 312 or the dialogue management component 310 may use a generative AI model 332 to improve their natural language understanding and dialogue management capabilities. For example, the NLU component 312 uses the generative AI model 332 to analyze the input of the user and extract relevant information, such as the intent of the user and entities. The dialogue management component 310 can also use the generative AI model 332 to identify and extract important information from unstructured text, such as a question of the user or a request. This can be done using methodologies such as named entity recognition, part-of-speech tagging, sentiment analysis, and the like. In some examples, the NLU component 312 communicates the prompt 302 directly to the generative AI model 332. The generative AI model 332 receives the prompt 302 and generates the chatbot response 308. In a similar manner, the dialogue management component 310 may receive the intent 320 and generate a generative AI model prompt that is communicated to the generative AI model 332. The generative AI model 332 receives the generative AI model prompt and generates the chatbot response 308.

[0073] FIG. 4A is a process flow diagram of a generative AI model prompting method and FIG. 4B is a collaboration diagram of the operations of a chatbot system, in accordance with some examples. A chatbot system, such as chatbot system 300 (of FIG. 3), uses a generative AI prompting method 400 to generate personality prompts and hint prompts for a generative AI model 332 (of FIG. 3).

[0074] In operation 402, the chatbot system 300 receives, via a user system 306 (of FIG. 3), a prompt 302 of a user 334 during an interactive session. For example, the user 334 uses user system 306 to access an interactive platform that

hosts the chatbot system 300. The user 334 enters a prompt 302 into the user system 306 and the user system 306 communicates the prompt 302 to the chatbot system 300. In some examples, the prompt 302 may include other types of data as well as text such as, but not limited to, image data, video data, audio data, electronic documents, links to data stored on the Internet or the user system 306, and the like. In addition, the prompt 302 may include media such as, but not limited to, audio media, image media, video media, textual media, and the like. Regardless of the data type of the prompt 302, keyword attribution and expansion may be used to automatically generate a cluster of keywords or attributes that are associated with the received prompt 302. For example, image recognition may be deployed to identify objects and location associated with visual media and image data and to generate a keyword cluster or cloud that is then associated with the image-based prompt.

[0075] The prompt 302 may furthermore be received through any number of interfaces and I/O components (e.g., the I/O components 208) of a user system 102. These include gesture-based inputs obtained from a biometric component and inputs received via a Brain-Computer Interface (BCI).

[0076] In some examples, the chatbot system 300 is integrated into various platforms such as, but not limited to, websites, messaging apps, and mobile apps, allowing users to interact with it through text or voice commands.

[0077] In operation 404, the chatbot system 300 determines an intent 320 (of FIG. 3) of the user 334 using the user prompt 302. For example, an NLU component 312 (of FIG. 3) receives the prompt 302 and any user feedback 316 including follow up messages from the user 334 and uses an intent processing pipeline to determine the intent 320. This pipeline uses Natural Language Processing (NLP) methodologies to map sets of conversations to a bag of intentful keywords and concepts. In addition to the keywords that are used in the original prompt 302, stemmed keywords and expanded concepts are also generated. The platform also assigns a weight to each concept based on the importance of those in the conversation. These keywords and concepts are aggregated and mapped to the user 334 as part of an intent profile or intent vector having weighted keywords and concepts based on the importance of those in the conversation.

[0078] In some examples, the chatbot system 300 associates a time factor to a keyword or concept where the time factor decays in time. For example, the chatbot system 300 attaches a time factor to the keywords and concepts, which indicates how fresh that concept should be used for targeting and bidding. For example, "Hotels in Cancun for spring break" vs "Planning for wedding next year" will have two different decaying factors. In some examples, the chatbot system 300 applies a time factor to a conversation state.

[0079] In some examples, an overall intent of the user is built using various signals or data such as user demographics, location, device, engagement with organic surfaces that are user interface and service features provided by the interactive server system of an interactive platform application and consumption patterns, and overall friend-graph proximity carrier signals or data that may be discernable from the user's affiliation with the interactive platform.

[0080] In some examples, the intent vector is an additional dimension that will be used to refresh and update an intent profile stored in a user profile 314 (of FIG. 3).

[0081] In some examples, the chatbot system 300 uses the user profile 314 that includes interactive platform user data 324 (of FIG. 3) of the user 334 to determine an intent of the user.

[0082] In some examples, the NLU component 312 (of FIG. 3) uses artificial intelligence methodologies including Machine Learning (ML) models to generate the intent 320. The chatbot system 300 uses the data on intent of the user and conversation context to improve its responses and optimization capabilities over time by ingesting the feedback. The chatbot system 300 collects engagement of the user 334 on ads, organic surfaces, and features of the interactive platforming system and also responses to the chatbot system 300 itself and feeds that into the NLU component 312. Accordingly, the chatbot system 300 can finetune or further pre-train the ML models of the NLU component 312 to not only take intent of the user into consideration, but also use follow up actions to finetune the intent of the user inference model.

[0083] In some examples, the chatbot system 300 collects a set of prompts during an interactive session. The chatbot system 300 maps the set of prompts to a set of keywords and/or concepts including an intent vector, as exemplified by

keywords and concepts. The chatbot system 300 assigns weights to the keywords of the set of keywords and/or concepts based on an importance score to the conversation of the keyword and/or concept, and determines the intent of the user based on the intent vector including the weighted keywords and/or concepts.

[0084] In some examples, the chatbot system 300 stores a conversation state in a user profile 314 as part of user data 324 of a series of interactive sessions so that the chatbot system 300 can have a context for conversations that occur over a plurality of interactive sessions.

[0085] In some examples, a knowledge base of the chatbot system 300 includes a set of information that the chatbot can use to understand and respond to an input of the user. This includes, but is not limited to, a predefined set of intents, entities, and responses, as well as external sources of information such as databases or APIs. In some examples, intent information of a friend, or friends, of the user 334 are used to understand, inform, and/or respond to a user's intent.

[0086] In some examples, an intent may not be determinable from the prompt of the user. For example, the user may make a declarative statement when no previous conversation history exists such as, but not limited to, making a declaration about the user's state, a declaration about a state of the user's immediate environment, or the like.

[0087] In operation 406, the chatbot system 300 determines a type of the prompt 302 using the prompt 302 and a conversation history stored in the user profile 314. For example, if the chatbot system 300 cannot determine any keywords in the prompt 302 that have relevance to the conversation history and none of the words in the prompt 302 have relevance to the chatbot system 300 or the interactive platform hosting the chatbot system 300, the chatbot system 300 determines a type of the prompt as being a personality prompt to be directly communicated to the generative AI model 332 (of FIG. 3) to prompt a response out of the generative AI model 332. However, if the chatbot system 300 determines one or more keywords in the prompt that have relevance to the conversation history, relevance to the chatbot, or relevance to the interactive platform hosting the chatbot, the chatbot system 300 determines a type of the prompt 302 as being an intent prompt to be used to generate an API call to one or more services outside of the chatbot system 300.

[0088] In response to determining that the prompt is a personality prompt, in operation 416, the chatbot system 300 communicates the prompt 302 as a generative AI model personality prompt 424 to the generative AI model 332.

[0089] In response to determining that the prompt 302 is an intent prompt from which a user intent can be determined, in operation 408, the chatbot system 300 generates one or more API calls 426 to one or more additional services 326 outside of the chatbot system 300. For example, if the chatbot system 300 determines that one or more keywords reference a physical location, the chatbot system 300 calls a mapping service passing the one or more keywords as arguments to the mapping service. In some examples, the chatbot system 300 determines that the keywords are related to a service or feature of the interactive platform hosting the chatbot system 300 and makes a call to that service or feature passing the one or more keywords as arguments to the service or feature.

[0090] In operation 410, the chatbot system 300 receives one or more return values 428 from the one or more additional services 326. For example, if the chatbot system 300 called a map service, the map service will return a physical location and information about the physical location. In some examples, if the chatbot system 300 called a service or feature of the interactive platform that hosts the chatbot system 300, the service or feature will return data generated by the service or information about the feature.

[0091] In operation 412, the chatbot system 300 generates a generative AI model hint prompt 422 using the one or more return values 428. For example, if the chatbot system 300 called a map service, the 300 uses the physical location and information about the physical location returned by the map service to generate the generative AI model hint prompt 422. In some examples, if the chatbot system 300 called a service or feature of the interactive platform that hosts the chatbot system 300, the chatbot system 300 uses the returned data generated by the service or returned information about the feature to generate the generative AI model hint prompt 422.

[0092] In operation 414, the chatbot system 300 communicates the generative AI model hint prompt 422 to the generative AI model 332.

[0093] In operation 418, the chatbot system 300 receives a chatbot response 308 from the generative AI model 332.

[0094] In operation 420, the chatbot system 300 provides the chatbot response 308 to the user 334 via the user system 306.

[0095] In some examples, the chatbot system 300 uses a dialogue management component 310 to generate a chatbot response 308 using an intent 320. For example, the dialogue management component 310 receives the intent 320 and communicates the intent 320 as a generative AI model prompt to the generative AI model 332. The generative AI model 332 receives the generative AI model prompt and generates the chatbot response 308. The generative AI model 332 communicates the chatbot response 308 to the dialogue management component 310. The dialogue management component 310 receives the chatbot response 308 and communicates the chatbot response 308 to the response filter component 304 for additional processing. In some examples, the dialogue management component 310 generates the chatbot response 308 using a set of additional services 326, such as, but not limited to, an image generation system. The dialogue management component 310 generates a service request 328 using the intent 320 and communicates the service request 328 to the additional service 326. The additional service 326 uses the service request 328 to generate the request response 330 and communicates the request response 330 to the dialogue management component 310. The dialogue management component 310 uses request response 330 to generate the chatbot response 308.

[0096] In some examples, the chatbot system 300 uses a response filter component 304 (of FIG. 3) to filter a raw response 318 generated by the dialogue management component 310. For example, the dialogue management component 310 communicates the raw response 318 to the response filter component 304 and the response filter component 304 filters the raw response 318 based on a set of filtering criteria to eliminate specified content from the raw response 318, for instance obscene words or concepts, or content that some may consider harmful. In some examples, the response filter component 304 generates an adjusted intent 322 based on filtering the raw response 318, and the dialogue management component 310 generates an additional raw response 318 using the adjusted intent 322.

[0097] In some examples, the generative AI model 332 and the additional service 326 are hosted by the same system that hosts other components of the

chatbot system 300. In some examples, the generative AI model 332 and the additional service 326 are hosted by a server system that is separate from the system that hosts the chatbot system 300, and the chatbot system 300 communicates with the generative AI model 332 and the additional service 326 over a network. For example, the chatbot system 300 receives the prompt 302 and communicates the prompt 302 to the generative AI model 332 residing on the separate server system. The generative AI model 332 receives the prompt 302 and generates a raw response 318. The generative AI model 332 then communicates the raw response 318 to the chatbot system 300. The chatbot system 300 receives the response for subsequent processing as described herein.

[0098] In some examples, as part of the chatbot response 308, the chatbot system 300 generates a set of personal AI system prompts that are displayed to the user 334 by the user system 306 that prompt the user 334 to interact with the chatbot system 300. The personal AI system prompts generated by the chatbot system 300 may include chat information such as, but not limited to, context sensitive material, instructions to the user 334, possible topics of conversation, and the like. In some examples, the chatbot system 300 uses the personal AI system prompts to suggest chat topics to a user 334 or to guide the user 334 through a conversation, such as providing instructional material on various topics. In some examples, the personal AI system prompts include suggestions of conversations or questions that are intended to solicit a user 334 to enter a prompt of the user 334. The suggested chats are aimed at helping the user 334 get to information they need, but provide a helpful side effect of generating additional user interactions with the chatbot system 300. These additional user interactions improve the ability of the chatbot system 300 determine user intent by provided additional context and information to the chatbot system 300.

[0099] In some examples, the system that hosts the chatbot system 300 may not be a component of an interactive platform but another system that provides services and information to a group of users such as, but not limited to, a platform that provides enterprise wide connectivity to a group of users such as employees of a company, clients of an enterprise provided professional services, educational institutions, and the like. In some of such examples, the content provided to the users may not be advertising, but may be other types of useful information such

as company policies, status messages for projects, newsworthy events, and the like.

[0100] FIG. 5 is an illustration of a chat session between a user and a chatbot, in accordance with some examples. To initiate the chat session 500, a user, such as user 334 (of FIG. 3), communicates an initial prompt 502 to a chatbot system, such as chatbot system 300 (of FIG. 3). At the initiation of the chat session 500, a conversation history 518 is an empty conversation history 508. The initial prompt 502 is a declarative statement by the user that is not relevant to the chatbot system 300, the conversation history 518, or the interactive platform hosting the chatbot system 300, so the chatbot system 300 makes an initial intent determination 504 of “none”. Accordingly, the chatbot system 300 communicates the initial prompt 502 to a generative AI model, such as generative AI model 332 (of FIG. 3) as a generative AI model personality prompt. The generative AI model 332 responds with a personality response 506 that prompts the user for an additional user prompt. The user responds with a prompt 510 that contains a keyword “Chinese” that is relevant to the conversation history with keywords 512 related to food. The chatbot system 300 determines a user intent, namely an intent to eat a particular kind of food. In response, the chatbot system 300 determines one or more API calls to additional services 514 such as a map service with arguments that were determined from the conversation history with keywords 512, a physical location of the user 334 as determined from a user profile and the interactive platform hosting the chatbot system 300. The chatbot system 300 receives return values from the additional services 326 (of FIG. 3) and generates a generative AI model hint prompt using the return values and communicates the generative AI model hint prompt to the generative AI model 332. The generative AI model 332 responds with a hint-based response 516 that indicates an establishment close to the user 334 where the user 334 can procure the type of food that they want.

[0101] FIG. 6A is a collaboration diagram of components of a chatbot system and FIG. 6B is an activity diagram of a chat memories method of a chatbot system, in accordance with some examples. A chatbot system 300 uses a chat memories method 600 to generate personality prompts in a form of augmented prompts generated during a current conversation between a user and a chatbot using stored past conversations.

[0102] Although a chat memories method 600 depicts a particular sequence of operations, the sequence may be altered without departing from the scope of the present disclosure. For example, some of the operations depicted may be performed in parallel or in a different sequence that does not materially affect the function of the routine. In other examples, different components of the chatbot system 300 (of FIG. 3) that implements the routine may perform functions at substantially the same time or in a specific sequence.

[0103] During one or more conversations between a user 334 and the chatbot system 300, the chatbot system 300 stores conversation histories 630 into a conversation history datastore 620. For example, the conversation histories 630 include sequences of user prompts 644, augmented prompts 640 and personalized responses 642 by generative AI model 332 during the one or more conversations.

[0104] Any conversation history 630 data collected by the chatbot system core 646 is captured and stored with only user approval and deleted on user request. Further, such conversation history 630 data may be used for very limited purposes, such as generating personalized responses 642. To ensure limited and authorized use of conversation history 630 data and other personally identifiable information (PII), access to this conversation history 630 data is restricted to authorized personnel only, if at all. Any use of conversation history 630 data may strictly be limited to the purpose of generation of the personalized responses 642, and the conversation history 630 data is not shared or sold to any third party without the explicit consent of the user. In addition, appropriate technical and organizational measures are implemented to ensure the security and confidentiality of this sensitive information.

[0105] In operation 602, the chatbot system 300 using a chatbot system core 646 such as, but not limited to, the NLU component 312 and the dialogue management component 310 (both of FIG. 3), retrieves a conversation history 628 from conversation history datastore at operation 602.

[0106] In operation 604, the chatbot system 300 generates summarized memories 632 using the conversation history 628. For example, the chatbot system 300 uses a summarizing component 616 to generate summarized memories 632 from the conversation history 628 using artificial intelligence methodologies and a summarizing model 648 that was previously generated using

machine learning methodologies. In some examples, the summarizing model 648 includes, but is not limited to, a generative AI model, a neural network, a learning vector quantization network, a logistic regression model, a support vector machine, a random decision forest, a naïve Bayes model, a linear discriminant analysis model, and a K-nearest neighbor model. In some examples, machine learning methodologies used to generate the summarizing model 648 may include, but are not limited to, supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, dimensionality reduction, self-learning, feature learning, sparse dictionary learning, and anomaly detection.

[0107] In operation 606, the chatbot system 300 generates moderated memories 634 using the summarized memories 632 and stores the memories 634 into a memories datastore 622. For example, the chatbot system 300 uses the moderating component 618 to eliminate specified content from the summarized memories 632, for instance obscene words or concepts, or content that some may consider harmful from the summarized memories 632 using artificial intelligence methodologies and a moderating model 650 that was previously generated using machine learning methodologies. In some examples, the moderating model 650 includes, but is not limited to, a generative AI model, a neural network, a learning vector quantization network, a logistic regression model, a support vector machine, a random decision forest, a naïve Bayes model, a linear discriminant analysis model, and a K-nearest neighbor model. In some examples, machine learning methodologies used to generate the moderating model 650 may include, but are not limited to, supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, dimensionality reduction, self-learning, feature learning, sparse dictionary learning, and anomaly detection.

[0108] In some examples, the chatbot system 300 also stores the memories 634 into a long term datastore 626 for later analysis.

[0109] In operation 608, the chatbot system 300 generates a current conversation context 638 using one or more user prompts and responses from the generative AI model 332 of a current conversation between the user 334 and the chatbot system 300. For example, the chatbot system 300 generates a conversation context 638 using a specified number of back-ward looking prompt and response interactions

of a current conversation. In some examples, the chatbot system 300 uses 5 previous interactions to generate the conversation context 638.

[0110] In operation 610, the chatbot system 300 generates context-appropriate memories 636 from the memories datastore 622 using the current conversation context 638. For example, the chatbot system 300 uses a retrieving component 624 to perform a semantic search of the memories datastore 622 for the context-appropriate memories 636. As an example conversation, a user may have a set of collected user memories:

- User likes to cook
- User has a dishwasher to fix
- User has a golden retriever
- User just got divorced

[0111] If the user's user prompt 644 is "Do you have any dog food recommendations?", without using the retrieving component 624, when these memories are used in an augmented prompt to a generative AI model, the generative AI model will try to inject all of the collected memories in a response, making the response large and disjointed, such as "I'm sorry to hear about your divorce. As for dog food recommendations, I suggest looking for high-quality brands with natural ingredients. It's important to consider your golden retriever's age, weight, and any specific health needs when choosing their food. Some popular brands include Blue Buffalo™, Wellness™, and Taste of the Wild™. Don't forget to check with your vet for any specific recommendations. And good luck with fixing your dishwasher! And let me know if you're interested in cooking something!"

[0112] However, the retrieving component 624 identifies the memory that the user has a golden retriever as the most relevant memory to the user prompt and uses that memory in an augmented prompt. This may result in a personalized response of "I'd be happy to help! When it comes to dog food, it's important to consider your golden retriever's age, weight, and any specific health needs. Some popular brands that offer high-quality dog food include Blue Buffalo™, Wellness™, and Taste of the Wild™. You could also consider a raw food diet or homemade dog food, but it's important to consult with your vet before making

any major changes. And don't forget to make sure your golden retriever has access to plenty of fresh water throughout the day!”

[0113] In operation 612, the chatbot system 300 generates an augmented prompt 640 using the context-appropriate memories 636 and the user prompt 644 and communicates the augmented prompt 640 to the generative AI model 332. For example, the chatbot system 300 uses the retrieving component 624 to generate the augmented prompt 640 using the user prompt 644, the context-appropriate memories 636, and an augmented prompt model 652 that was previously generated using machine learning methodologies. In some examples, the augmented prompt model 652 includes, but is not limited to, a generative AI model, a neural network, a learning vector quantization network, a logistic regression model, a support vector machine, a random decision forest, a naïve Bayes model, a linear discriminant analysis model, and a K-nearest neighbor model. In some examples, machine learning methodologies used to generate the augmented prompt model 652 may include, but are not limited to, supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, dimensionality reduction, self-learning, feature learning, sparse dictionary learning, and anomaly detection.

[0114] In some examples, contextual matching retrieval is performed via prompting of the generative AI model. For example, the chatbot system 300 instructs the generative AI model 332 to only use relevant user knowledge when generating a personalized response. In an example, the chatbot system 300 adds prompting such as “You only use contextual information relevant to the question” to the augmented prompt 640. In some examples, a length limitation is added to the contextual information to mitigate computation costs.

[0115] In some examples, the retrieving component 624 retrieves relevant information from all stored user knowledge, and only sends relevant contexts to the generative AI model 332. Instead of instructing the generative AI model 332 to do the filtering. In some examples, the retrieving component 624 determines contextual important memories in the memories datastore 622 using a semantic search process.

[0116] In some examples, the memories datastore 622 is a Key-Value data store. The key is a user id, and the value is the summarized memories 634. During

operation, the retrieved context-appropriate memories 636 will be cached throughout the session. In some examples, instead of storing raw text directly, precomputed embeddings of raw texts for stored.

[0117] In some examples, in the augmented prompt 640, the generative AI model 332 is instructed to output responses based on personalized context. The personalized context includes the context-appropriate memories 636 retrieved from the memories datastore 622. Below is an example schema for augmented prompt:

```
<user utterance>
```

```
<blank line>
```

```
You write the response with help from information  
below. You only use information relevant to the  
question. You only use non-negative information.
```

```
<memory>
```

[0118] In some examples, the generative AI model 332 is instructed to only use positive memory items from the context-appropriate memories 636, to avoid potential negative user experiences brought on by negative memories. In some examples, both positive and negative memories are stored in the memories datastore 622 allowing the generative AI model 332 to generate empathetic personalized response 642 in database.

[0119] In some examples, the generative AI model 332 is prompted by an augmented prompt such as, but not limited to: "Your goal is to summarize user interests, hobbies, long-standing facts and goals from user messages and previous summary of the user. If user mentions their friends' or family members' interests, hobbies, long-standing facts and goals, you also summarize it. Previous summary only represents information in the past. Update previous summary if some information is outdated. Otherwise, append previous summary to current summary. You only summarize user's messages. You don't summarize for AI or user's conversation partner. Your response is "None" if not enough information about user's interests, hobbies, long-standing facts and goals. You don't need to answer user's question. Your response is "None" if user is asking for answers for homework or coursework."

[0120] In some examples, the memories datastore 622 is regularly updated to reflect changes in the life of the user 334.

[0121] In operation 614, the chatbot system 300 receives a personalized response 642 from the generative AI model 332 and provides the personalized response 642 to the user 334 during the current conversation between the user 334 and the chatbot system 300.

[0122] In some examples, through the use of the memories datastore 622, the chatbot system 300 has a better user understanding by leveraging users' conversation history by internalizing and persisting the information via a user knowledge database.

[0123] In some examples, the chatbot system 300 personalizes responses to users in appropriate conversations and using appropriate user contextual information as the chatbot system 300 may avoid scenarios where the personalized response 642 includes too much user information.

[0124] In some examples, the chatbot system 300 can scale up to longer conversations and time windows. If users mentioned important factual information several months ago, the chatbot system 300 is still able to retrieve it and use it to personalize the responses whenever needed.

[0125] In some examples, when the chatbot system 300 receives a request from the user 334 to delete a conversation, the chatbot system 300 deletes the conversation and the memories stored in the memories datastore 622 associated with that conversation as well.

[0126] In some examples, the retrieved context-appropriate memories 636 are only based on each user's provided information. The information is not shared across users.

[0127] In some examples, the memories datastore 622 is isolated for each conversation group and only memories generated from respective historical chats are used to generate respective augmented prompts associated with a conversation group. For example, historical chats between the chatbot system 300 and user_B will not be used to augment chats with user_A. In some examples, an exception is made if user_A mentioned some fact about user_B to the chatbot system, then the chatbot system 300 uses this information when user_A asks questions about user_B later. In some examples, the chatbot system 300 will not search a

memories datastore of user_B to generate a personalized response for user_A's user prompt.

[0128] In some examples, the generative AI model 332 and the chatbot system 300 are hosted by the same system. In some examples, the generative AI model 332 is hosted by a server system that is separate from the chatbot system 300, and the chatbot system 300 communicates with the generative AI model 332 over a network.

MACHINE-LEARNING PIPELINE

[0129] FIG. 7 illustrates a machine-learning pipeline 800 and FIG. 8 is a flowchart depicting a method of the machine-learning pipeline 800, according to some examples. The machine-learning pipeline 800 may be used to generate a trained machine-learning program 802 of FIG. 8, to perform operations associated with searches and query responses. In some examples, the trained machine-learning program 802 includes a generative AI model such as, but not limited to a generative AI model 332 of FIG. 3, a summarizing model 648 of FIG. 6A, a moderating model 650 of FIG. 6A, an augmented prompt model 652 of FIG. 6A, and the like.

Overview

[0130] Broadly, machine learning may involve using computer algorithms to automatically learn patterns and relationships in data, potentially without the need for explicit programming. Machine learning algorithms can be divided into three main 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 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 include clustering, principal component analysis, and generative models like 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.

[0131] 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 another 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. 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 types of machine learning algorithms include decision trees, k-nearest neighbors, clustering algorithms, and deep learning algorithms such as convolutional neural networks (CNN), recurrent neural networks (RNN), and transformer models. The choice of algorithm depends on the nature of the data, the complexity of the problem, and the performance requirements of the application.

[0132] The performance of machine learning models is typically 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.

[0133] Although several specific examples of machine learning algorithms are discussed herein, the principles discussed herein can be applied to other machine learning algorithms as well. Deep learning algorithms such as convolutional neural networks, recurrent neural networks, and transformers, as well as more

traditional machine learning algorithms like decision trees, random forests, and gradient boosting may be used in various machine learning applications.

[0134] Three example types of problems in machine learning are classification problems, regression problems, and generation 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). Generation algorithms aim at producing new examples that are similar to examples provided for training. For instance, a text generation algorithm is trained on many text documents and is configured to generate new coherent text with similar statistical properties as the training data.

Training phases 804

[0135] Generating a trained machine-learning program 802 may include multiple phases that form part of the machine-learning pipeline 800, including for example the following phases illustrated in FIG. 7:

- 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.
- Feature engineering 704 : This phase may include selecting and transforming the training data 806 to create features that are useful for predicting the target variable. Feature engineering may include (1) receiving features 808 (e.g., as structured or labeled data in supervised learning) and/or (2) identifying features 808 (e.g., unstructured or unlabeled data for unsupervised learning) in training data 806.
- Model selection and training 706 : This phase may 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.
- Model evaluation 708 : This phase may include evaluating the performance of a trained model (e.g., the trained machine-learning program 802) 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.

- Prediction 710 : This phase involves using a trained model (e.g., trained machine-learning program 802) to generate predictions on new, unseen data.
- Validation, refinement or retraining 712 : This phase may include updating a model based on feedback generated from the prediction phase, such as new data or user feedback.
- Deployment 714 : This phase may include integrating the trained model (e.g., the trained machine-learning program 802) into a more extensive system or application, such as a web service, mobile app, or IoT device. This phase can involve setting up APIs, building a user interface, and ensuring that the model is scalable and can handle large volumes of data.

[0136] FIG. 8 illustrates further details of two example phases, namely a training phase 804 (e.g., part of the model selection and trainings 706) and a prediction phase 810 (part of prediction 710). Prior to the training phase 804, feature engineering 704 is used to identify features 808. This may include identifying informative, discriminating, and independent features for effectively operating the trained machine-learning program 802 in pattern recognition, classification, and regression. In some examples, the training data 806 includes labeled data, known for pre-identified features 808 and one or more outcomes. Each of the features 808 may be a variable or attribute, such as an individual measurable property of a process, article, system, or phenomenon represented by a data set (e.g., the training data 806). Features 808 may also be of different types, such as numeric features, strings, and graphs, and may include one or more of content 812, concepts 814, attributes 816, historical data 818, and/or user data 820, merely for example.

[0137] In training phase 804, the machine-learning pipeline 800 uses the training data 806 to find correlations among the features 808 that affect a predicted outcome or prediction/inference data 822.

[0138] With the training data 806 and the identified features 808, the trained machine-learning program 802 is trained during the training phase 804 during machine-learning program training 824. The machine-learning program training 824 appraises values of the features 808 as they correlate to the training data 806.

The result of the training is the trained machine-learning program 802 (e.g., a trained or learned model).

[0139] Further, the training phase 804 may involve machine learning, in which the training data 806 is structured (e.g., labeled during preprocessing operations). The trained machine-learning program 802 implements a neural network 826 capable of performing, for example, classification and clustering operations. In other examples, the training phase 804 may involve deep learning, in which the training data 806 is unstructured, and the trained machine-learning program 802 implements a deep neural network 826 that can perform both feature extraction and classification/clustering operations.

[0140] In some examples, a neural network 826 may be generated during the training phase 804, and implemented within the trained machine-learning program 802. The neural network 826 includes a hierarchical (e.g., layered) organization of neurons, with each layer consisting of multiple neurons or nodes. Neurons in the input layer receive the input data, while neurons in the output layer produce the final output of the network. Between the input and output layers, there may be one or more hidden layers, each consisting of multiple neurons.

[0141] Each neuron in the neural network 826 operationally computes a function, such as an activation function, which takes as input the weighted sum of the outputs of the neurons in the previous layer, as well as a bias term. The output of this function is then passed as input to the neurons in the next layer. If the output of the activation function exceeds a certain threshold, an output is communicated from that neuron (e.g., transmitting neuron) to a connected neuron (e.g., receiving neuron) in successive layers. The connections between neurons have associated weights, which define the influence of the input from a transmitting neuron to a receiving neuron. During the training phase, these weights are adjusted by the learning algorithm to optimize the performance of the network. Different types of neural networks may use different activation functions and learning algorithms, affecting their performance on different tasks. The layered organization of neurons and the use of activation functions and weights enable neural networks to model complex relationships between inputs and outputs, and to generalize to new inputs that were not seen during training.

[0142] In some examples, the neural network 826 may also be one of several different types of neural networks, such as a single-layer feed-forward network, a Multilayer Perceptron (MLP), an Artificial Neural Network (ANN), a Recurrent Neural Network (RNN), a Long Short-Term Memory Network (LSTM), a Bidirectional Neural Network, a symmetrically connected neural network, a Deep Belief Network (DBN), a Convolutional Neural Network (CNN), a Generative Adversarial Network (GAN), an Autoencoder Neural Network (AE), a Restricted Boltzmann Machine (RBM), a Hopfield Network, a Self-Organizing Map (SOM), a Radial Basis Function Network (RBFN), a Spiking Neural Network (SNN), a Liquid State Machine (LSM), an Echo State Network (ESN), a Neural Turing Machine (NTM), or a Transformer Network, merely for example.

[0143] In addition to the training phase 804, a validation phase may be performed on a separate dataset known as the validation dataset. The validation dataset is used to tune the hyperparameters of a model, such as the learning rate and the regularization parameter. The hyperparameters are adjusted to improve the model's performance on the validation dataset.

[0144] Once a model is fully trained and validated, in a testing phase, the model may be tested on a new dataset. The testing dataset is used to evaluate the model's performance and ensure that the model has not overfitted the training data.

[0145] In prediction phase 810, the trained machine-learning program 802 uses the features 808 for analyzing query data 828 to generate inferences, outcomes, or predictions, as examples of a prediction/inference data 822. For example, during prediction phase 810, the trained machine-learning program 802 generates an output. Query data 828 is provided as an input to the trained machine-learning program 802, and the trained machine-learning program 802 generates the prediction/inference data 822 as output, responsive to receipt of the query data 828.

[0146] In some examples, the trained machine-learning program 802 may be a generative AI model. Generative AI is a term that may refer to any type of artificial intelligence that can create new content from training data 806. For example, generative AI can produce text, images, video, audio, code, or synthetic data similar to the original data but not identical.

[0147] Some of the techniques that may be used in generative AI are:

- Convolutional Neural Networks (CNNs): CNNs may be used for image recognition and computer vision tasks. CNNs may, for example, be designed to extract features from images by using filters or kernels that scan the input image and highlight important patterns.
- Recurrent Neural Networks (RNNs): RNNs may be used for processing sequential data, such as speech, text, and time series data, for example. RNNs employ feedback loops that allow them to capture temporal dependencies and remember past inputs.
- Generative adversarial networks (GANs): GANs may include two neural networks: a generator and a discriminator. The generator network attempts to create realistic content that can “fool” the discriminator network, while the discriminator network attempts to distinguish between real and fake content. The generator and discriminator networks compete with each other and improve over time.
- Variational autoencoders (VAEs): VAEs may encode input data into a latent space (e.g., a compressed representation) and then decode it back into output data. The latent space can be manipulated to generate new variations of the output data. VAEs may use self-attention mechanisms to process input data, allowing them to handle long text sequences and capture complex dependencies.
- Transformer models: Transformer models may use attention mechanisms to learn the relationships between different parts of input data (such as words or pixels) and generate output data based on these relationships. Transformer models can handle sequential data, such as text or speech, as well as non-sequential data, such as images or code.

[0148] In generative AI examples, the query data 828 may include text, audio, image, video, numeric, or media content prompts and the output prediction/inference data 822 includes text, images, video, audio, code, or synthetic data.

DATA ARCHITECTURE

[0149] FIG. 9 is a schematic diagram illustrating data structures 900, which may be stored in the database 904 of the interactive server system 110 (of FIG. 1), according to some examples. While the content of the database 904 is shown to

include multiple tables, it will be appreciated that the data could be stored in other types of data structures (e.g., as an object-oriented database).

[0150] Any data collected by an interactive platform is captured and stored with only user approval and deleted on user request. Further, such data may be used for very limited purposes, such as identity verification. To ensure limited and authorized use of data and other personally identifiable information, access to this data is restricted to authorized personnel only, if at all. Any use of data may strictly be limited to a designated purpose and the data is not shared or sold to any third party without the explicit consent of the user. In addition, appropriate technical and organizational measures are implemented to ensure the security and confidentiality of this sensitive information.

[0151] The database 904 includes message data stored within a message table 906. This message data includes, for any particular message, at least message sender data, message recipient (or receiver) data, and a payload. Further details regarding information that may be included in a message, and included within the message data stored in the message table 906, are described below with reference to FIG. 10.

[0152] An entity table 908 stores entity data, and is linked (e.g., referentially) to an entity graph 910 and profile data 902. Entities for which records are maintained within the entity table 908 may include individuals, corporate entities, organizations, objects, places, events, and so forth. Regardless of entity type, any entity regarding which the interactive server system 110 stores data may be a recognized entity. Each entity is provided with a unique identifier, as well as an entity type identifier (not shown).

[0153] The entity graph 910 stores information regarding relationships and associations between entities. Such relationships may be social, professional (e.g., work at a common corporation or organization), interest-based, or activity-based, merely for example. Certain relationships between entities may be unidirectional, such as a subscription by an individual user to digital content of a commercial or publishing user (e.g., a newspaper or other digital media outlet, or a brand). Other relationships may be bidirectional, such as a “friend” relationship between individual users of the interactive platform 100 (of FIG. 1).

[0154] Certain permissions and relationships may be attached to each relationship, and also to each direction of a relationship. For example, a bidirectional relationship (e.g., a friend relationship between individual users) may include authorization for the publication of digital content items between the individual users, but may impose certain restrictions or filters on the publication of such digital content items (e.g., based on content characteristics, location data or time of day data). Similarly, a subscription relationship between an individual user and a commercial user may impose different degrees of restrictions on the publication of digital content from the commercial user to the individual user, and may significantly restrict or block the publication of digital content from the individual user to the commercial user. A particular user, as an example of an entity, may record certain restrictions (e.g., by way of privacy settings) in a record for that entity within the entity table 908. Such privacy settings may be applied to all types of relationships within the context of the interactive platform 100, or may selectively be applied to only certain types of relationships.

[0155] The profile data 902 stores multiple types of profile data about a particular entity. The profile data 902 may be selectively used and presented to other users of the interactive platform 100 based on privacy settings specified by a particular entity. Where the entity is an individual, the profile data 902 includes, for example, a username, telephone number, address, settings (e.g., notification and privacy settings), as well as a user-selected avatar representation (or collection of such avatar representations). A particular user may then selectively include one or more of these avatar representations within the content of messages communicated via the interactive platform 100, and on map interfaces displayed by interaction clients 104 to other users. The collection of avatar representations may include “status avatars,” which present a graphical representation of a status or activity that the user may select to communicate at a particular time.

[0156] Where the entity is a group, the profile data 902 for the group may similarly include one or more avatar representations associated with the group, in addition to the group name, members, and various settings (e.g., notifications) for the relevant group.

[0157] The database 904 also stores augmentation data, such as overlays or filters, in an augmentation table 912. The augmentation data is associated with and applied to videos (for which data is stored in a video table 914) and images (for which data is stored in an image table 916).

[0158] Filters, in some examples, are overlays that are displayed as overlaid on an image or video during presentation to a message receiver. Filters may be of various types, including user-selected filters from a set of filters presented to a message sender by the interaction client 104 when the message sender is composing a message. Other types of filters include geolocation filters (also known as geo-filters), which may be presented to a message sender based on geographic location. For example, geolocation filters specific to a neighborhood or special location may be presented within a user interface by the interaction client 104, based on geolocation information determined by a Global Positioning System (GPS) unit of the user system 102.

[0159] Another type of filter is a data filter, which may be selectively presented to a message sender by the interaction client 104 based on other inputs or information gathered by the user system 102 during the message creation process. Examples of data filters include current temperature at a specific location, a current speed at which a message sender is traveling, battery life for a user system 102, or the current time.

[0160] Other augmentation data that may be stored within the image table 916 includes augmented reality content items (e.g., corresponding to applying Lenses or augmented reality experiences). An augmented reality content item may be a real-time special effect and sound that may be added to an image or a video.

[0161] As described above, augmentation data includes augmented reality (AR), virtual reality (VR) and mixed reality (MR) content items, overlays, image transformations, images, and modifications that may be applied to image data (e.g., videos or images). This includes real-time modifications, which modify an image as it is captured using device sensors (e.g., one or multiple cameras) of the user system 102 and then displayed on a screen of the user system 102 with the modifications. This also includes modifications to stored content, such as video clips in a collection or group that may be modified. For example, in a user system 102 with access to multiple augmented reality content items, a user can use a

single video clip with multiple augmented reality content items to see how the different augmented reality content items will modify the stored clip. Similarly, real-time video capture may use modifications to show how video images currently being captured by sensors of a user system 102 would modify the captured data. Such data may simply be displayed on the screen and not stored in memory, or the content captured by the device sensors may be recorded and stored in memory with or without the modifications (or both). In some systems, a preview feature can show how different augmented reality content items will look within different windows in a display at the same time. This can, for example, enable multiple windows with different pseudorandom animations to be viewed on a display at the same time.

[0162] Data and various systems using augmented reality content items or other such transform systems to modify content using this data can thus involve detection of objects (e.g., faces, hands, bodies, cats, dogs, surfaces, objects, etc.), tracking of such objects as they leave, enter, and move around the field of view in video frames, and the modification or transformation of such objects as they are tracked. In various examples, different methods for achieving such transformations may be used. Some examples may involve generating a three-dimensional mesh model of the object or objects, and using transformations and animated textures of the model within the video to achieve the transformation. In some examples, tracking of points on an object may be used to place an image or texture (which may be two-dimensional or three-dimensional) at the tracked position. In still further examples, neural network analysis of video frames may be used to place images, models, or textures in content (e.g., images or frames of video). Augmented reality content items thus refer both to the images, models, and textures used to create transformations in content, as well as to additional modeling and analysis information needed to achieve such transformations with object detection, tracking, and placement.

[0163] Real-time video processing can be performed with any kind of video data (e.g., video streams, video files, etc.) saved in a memory of a computerized system of any kind. For example, a user can load video files and save them in a memory of a device, or can generate a video stream using sensors of the device. Additionally, any objects can be processed using a computer animation model,

such as a human's face and parts of a human body, animals, or non-living things such as chairs, cars, or other objects.

[0164] In some examples, when a particular modification is selected along with content to be transformed, elements to be transformed are identified by the computing device, and then detected and tracked if they are present in the frames of the video. The elements of the object are modified according to the request for modification, thus transforming the frames of the video stream. Transformation of frames of a video stream can be performed by different methods for different kinds of transformation. For example, for transformations of frames mostly referring to changing forms of an object's elements, characteristic points for each element of an object are calculated (e.g., using an Active Shape Model (ASM) or other known methods). Then, a mesh based on the characteristic points is generated for each element of the object. This mesh is used in the following stage of tracking the elements of the object in the video stream. In the process of tracking, the mesh for each element is aligned with a position of each element. Then, additional points are generated on the mesh.

[0165] In some examples, transformations changing some areas of an object using its elements can be performed by calculating characteristic points for each element of an object and generating a mesh based on the calculated characteristic points. Points are generated on the mesh, and then various areas based on the points are generated. The elements of the object are then tracked by aligning the area for each element with a position for each of the at least one element, and properties of the areas can be modified based on the request for modification, thus transforming the frames of the video stream. Depending on the specific request for modification, properties of the mentioned areas can be transformed in different ways. Such modifications may involve changing the color of areas; removing some part of areas from the frames of the video stream; including new objects into areas that are based on a request for modification; and modifying or distorting the elements of an area or object. In various examples, any combination of such modifications or other similar modifications may be used. For certain models to be animated, some characteristic points can be selected as control points to be used in determining the entire state-space of options for the model animation.

[0166] In some examples of a computer animation model to transform image data using face detection, the face is detected on an image using a specific face detection algorithm (e.g., Viola-Jones). Then, an Active Shape Model (ASM) algorithm is applied to the face region of an image to detect facial feature reference points.

[0167] Other methods and algorithms suitable for face detection can be used. For example, in some examples, features are located using a landmark, which represents a distinguishable point present in most of the images under consideration. For facial landmarks, for example, the location of the left eye pupil may be used. If an initial landmark is not identifiable (e.g., if a person has an eyepatch), secondary landmarks may be used. Such landmark identification procedures may be used for any such objects. In some examples, a set of landmarks forms a shape. Shapes can be represented as vectors using the coordinates of the points in the shape. One shape is aligned to another with a similarity transform (allowing translation, scaling, and rotation) that minimizes the average Euclidean distance between shape points. The mean shape is the mean of the aligned training shapes.

[0168] A transformation system can capture an image or video stream on a user device (e.g., the user system 102) and perform complex image manipulations locally on the user system 102 while maintaining a suitable user experience, computation time, and power consumption. The complex image manipulations may include size and shape changes, emotion transfers (e.g., changing a face from a frown to a smile), state transfers (e.g., aging a subject, reducing apparent age, changing gender), style transfers, graphical element application, and any other suitable image or video manipulation implemented by a convolutional neural network that has been configured to execute efficiently on the user system 102.

[0169] In some examples, a computer animation model to transform image data can be used by a system where a user may capture an image or video stream of the user (e.g., a selfie) using the user system 102 having a neural network operating as part of an interaction client 104 operating on the user system 102. The transformation system operating within the interaction client 104 determines the presence of a face within the image or video stream and provides modification icons associated with a computer animation model to transform image data, or the

computer animation model can be present as associated with an interface described herein. The modification icons include changes that are the basis for modifying the user's face within the image or video stream as part of the modification operation. Once a modification icon is selected, the transform system initiates a process to convert the image of the user to reflect the selected modification icon (e.g., generate a smiling face on the user). A modified image or video stream may be presented in a graphical user interface displayed on the user system 102 as soon as the image or video stream is captured and a specified modification is selected. The transformation system may implement a complex convolutional neural network on a portion of the image or video stream to generate and apply the selected modification. That is, the user may capture the image or video stream and be presented with a modified result in real-time or near real-time once a modification icon has been selected. Further, the modification may be persistent while the video stream is being captured, and the selected modification icon remains toggled. Machine-taught neural networks may be used to enable such modifications.

[0170] The graphical user interface, presenting the modification performed by the transform system, may supply the user with additional interaction options. Such options may be based on the interface used to initiate the content capture and selection of a particular computer animation model (e.g., initiation from a content creator user interface). In various examples, a modification may be persistent after an initial selection of a modification icon. The user may toggle the modification on or off by tapping or otherwise selecting the face being modified by the transformation system and store it for later viewing or browsing to other areas of the imaging application. Where multiple faces are modified by the transformation system, the user may toggle the modification on or off globally by tapping or selecting a single face modified and displayed within a graphical user interface. In some examples, individual faces, among a group of multiple faces, may be individually modified, or such modifications may be individually toggled by tapping or selecting the individual face or a series of individual faces displayed within the graphical user interface.

[0171] A story table 918 stores data regarding collections of messages and associated image, video, or audio data, which are compiled into a collection (e.g., a story or a gallery). The creation of a particular collection may be initiated by a

particular user (e.g., each user for which a record is maintained in the entity table 908). A user may create a “personal story” in the form of a collection of content that has been created and sent/broadcast by that user. To this end, the user interface of the interaction client 104 may include an icon that is user-selectable to enable a message sender to add specific content to his or her personal story.

[0172] A collection may also constitute a “live story,” which is a collection of content from multiple users that is created manually, automatically, or using a combination of manual and automatic methodologies. For example, a “live story” may constitute a curated stream of user-submitted content from various locations and events. Users whose user devices have location services enabled and are at a common location event at a particular time may, for example, be presented with an option, via a user interface of the interaction client 104, to contribute content to a particular live story. The live story may be identified to the user by the interaction client 104, based on his or her location. The end result is a “live story” told from a community perspective.

[0173] A further type of content collection is known as a “location story,” which enables a user whose user system 102 is located within a specific geographic location (e.g., on a college or university campus) to contribute to a particular collection. In some examples, a contribution to a location story may require a second degree of authentication to verify that the end-user belongs to a specific organization or other entity (e.g., is a student on the university campus).

[0174] As mentioned above, the video table 914 stores video data that, in some examples, is associated with messages for which records are maintained within the message table 906. Similarly, the image table 916 stores image data associated with messages for which message data is stored in the entity table 908. The entity table 908 may associate various augmentations from the augmentation table 912 with various images and videos stored in the image table 916 and the video table 914.

[0175] The databases 904 also include social network information collected by an interactive platform. The social network information may include without limitation relationship and communication data for users of the interactive platform. The social network information can be used to group two or more users and offer additional functionality of the interactive platform 100. Examples of

relationships include, but are not limited to, best friends relationships where two or more users are determined to be mutual best friends based on a frequency of their interactions, users who have common interests in current events, users who share an affiliation through social clubs or philanthropic organizations, and the like. Examples of communications include without limitation chats, private and public messages, exchanges of media such as images, videos, audio recordings, and the like.

DATA COMMUNICATIONS ARCHITECTURE

[0176] FIG. 10 is a schematic diagram illustrating a structure of a message 1000, according to some examples, generated by an interaction client 104 for communication to a further interaction client 104 via the Interactive servers 124. The content of a particular message 1000 is used to populate the message table 906 stored within the database 904, accessible by the Interactive servers 124. Similarly, the content of a message 1000 is stored in memory as “in-transit” or “in-flight” data of the user system 102 or the Interactive servers 124. A message 1000 is shown to include the following example components:

- Message identifier 1002 : a unique identifier that identifies the message 1000.
- Message text payload 1034 : text, to be generated by a user via a user interface of the user system 102, and that is included in the message 1000.
- Message image payload 1004 : image data, captured by a camera component of a user system 102 or retrieved from a memory component of a user system 102, and that is included in the message 1000. Image data for a sent or received message 1000 may be stored in the image table 1006.
- Message video payload 1008 : video data, captured by a camera component or retrieved from a memory component of the user system 102, and that is included in the message 1000. Video data for a sent or received message 1000 may be stored in the video table 1010.
- Message audio payload 1012 : audio data, captured by a microphone or retrieved from a memory component of the user system 102, and that is included in the message 1000.
- Message augmentation data 1014 : augmentation data (e.g., filters, stickers, or other annotations or enhancements) that represents augmentations

to be applied to message image payload 1004, message video payload 1008, or message audio payload 1012 of the message 1000. Augmentation data for a sent or received message 1000 may be stored in the augmentation table 1016.

- Message duration parameter 1018 : parameter value indicating, in seconds, the amount of time for which content of the message (e.g., the message image payload 1004, message video payload 1008, message audio payload 1012) is to be presented or made accessible to a user via the interaction client 104.
- Message geolocation parameter 1020 : geolocation data (e.g., latitudinal and longitudinal coordinates) associated with the content payload of the message. Multiple message geolocation parameter 1020 values may be included in the payload, each of these parameter values being associated with respect to content items included in the content (e.g., a specific image within the message image payload 1004, or a specific video in the message video payload 1008).
- Message story identifier 1022 : identifier values identifying one or more content collections (e.g., “stories” identified in the story table 1024) with which a particular content item in the message image payload 1004 of the message 1000 is associated. For example, multiple images within the message image payload 1004 may each be associated with multiple content collections using identifier values.
- Message tag 1026: each message 1000 may be tagged with multiple tags, each of which is indicative of the subject matter of content included in the message payload. For example, where a particular image included in the message image payload 1004 depicts an animal (e.g., a lion), a tag value may be included within the message tag 1026 that is indicative of the relevant animal. Tag values may be generated manually, based on user input, or may be automatically generated using, for example, image recognition.
- Message sender identifier 1028: an identifier (e.g., a messaging system identifier, email address, or device identifier) indicative of a user of the user system 102 on which the message 1000 was generated and from which the message 1000 was sent.
- Message receiver identifier 1030: an identifier (e.g., a messaging system identifier, email address, or device identifier) indicative of a user of the user system 102 to which the message 1000 is addressed.

[0177] The contents (e.g., values) of the various components of message 1000 may be pointers to locations in tables within which content data values are stored. For example, an image value in the message image payload 1004 may be a pointer to (or address of) a location within an image table 1006. Similarly, values within the message video payload 1008 may point to data stored within a video table 1010, values stored within the message augmentation data 1014 may point to data stored in an augmentation table 1016, values stored within the message story identifier 1022 may point to data stored in a story table 1024, and values stored within the message sender identifier 1028 and the message receiver identifier 1030 may point to user records stored within an entity table 1032.

[0178] FIG. 11 is a schematic diagram illustrating an access-limiting process 1100, in terms of which access to content (e.g., an ephemeral message 1102 and associated multimedia payload of data) or a content collection (e.g., an ephemeral message group 1104) may be time-limited (e.g., made ephemeral).

[0179] An ephemeral message 1102 is shown to be associated with a message duration parameter 1106, the value of which determines the amount of time that the ephemeral message 1102 will be displayed to a receiving user of the ephemeral message 1102 by the interaction client 104. In some examples, an ephemeral message 1102 is viewable by a receiving user for up to a maximum of 10 seconds, depending on the amount of time that the message sender specifies using the message duration parameter 1106.

[0180] The message duration parameter 1106 and the message receiver identifier 1108 are shown to be inputs to a message timer 1110, which is responsible for determining the amount of time that the ephemeral message 1102 is shown to a particular receiving user identified by the message receiver identifier 1108. In particular, the ephemeral message 1102 will be shown to the relevant receiving user for a time period determined by the value of the message duration parameter 1106. The message timer 1110 is shown to provide output to a more generalized messaging system 1112, which is responsible for the overall timing of display of content (e.g., an ephemeral message 1102) to a receiving user.

[0181] The ephemeral message 1102 is shown in FIG. 11 to be included within an ephemeral message group 1104 (e.g., a collection of messages in a personal story, or an event story). The ephemeral message group 1104 has an associated

group duration parameter 1114, a value of which determines a time duration for which the ephemeral message group 1104 is presented and accessible to users of the interactive platform 100. The group duration parameter 1114, for example, may be the duration of a music concert, where the ephemeral message group 1104 is a collection of content pertaining to that concert. Alternatively, a user (either the owning user or a curator user) may specify the value for the group duration parameter 1114 when performing the setup and creation of the ephemeral message group 1104.

[0182] Additionally, each ephemeral message 1102 within the ephemeral message group 1104 has an associated group participation parameter 1116, a value of which determines the duration of time for which the ephemeral message 1102 will be accessible within the context of the ephemeral message group 1104. Accordingly, a particular ephemeral message group 1104 may “expire” and become inaccessible within the context of the ephemeral message group 1104 prior to the ephemeral message group 1104 itself expiring in terms of the group duration parameter 1114. The group duration parameter 1114, group participation parameter 1116, and message receiver identifier 1108 each provide input to a group timer 1118, which operationally determines, firstly, whether a particular ephemeral message 1102 of the ephemeral message group 1104 will be displayed to a particular receiving user and, if so, for how long. Note that the ephemeral message group 1104 is also aware of the identity of the particular receiving user as a result of the message receiver identifier 1108.

[0183] Accordingly, the group timer 1118 operationally controls the overall lifespan of an associated ephemeral message group 1104 as well as an individual ephemeral message 1102 included in the ephemeral message group 1104. In some examples, each and every ephemeral message 1102 within the ephemeral message group 1104 remains viewable and accessible for a time period specified by the group duration parameter 1114. In a further example, a certain ephemeral message 1102 may expire within the context of ephemeral message group 1104 based on a group participation parameter 1116. Note that a message duration parameter 1106 may still determine the duration of time for which a particular ephemeral message 1102 is displayed to a receiving user, even within the context of the ephemeral message group 1104. Accordingly, the message duration parameter 1106 determines the duration of time that a particular ephemeral

message 1102 is displayed to a receiving user regardless of whether the receiving user is viewing that ephemeral message 1102 inside or outside the context of an ephemeral message group 1104.

[0184] The messaging system 1112 may furthermore operationally remove a particular ephemeral message 1102 from the ephemeral message group 1104 based on a determination that it has exceeded an associated group participation parameter 1116. For example, when a message sender has established a group participation parameter 1116 of 24 hours from posting, the messaging system 1112 will remove the relevant ephemeral message 1102 from the ephemeral message group 1104 after the specified 24 hours. The messaging system 1112 also operates to remove an ephemeral message group 1104 when either the group participation parameter 1116 for each and every ephemeral message 1102 within the ephemeral message group 1104 has expired, or when the ephemeral message group 1104 itself has expired in terms of the group duration parameter 1114.

[0185] In certain use cases, a creator of a particular ephemeral message group 1104 may specify an indefinite group duration parameter 1114. In this case, the expiration of the group participation parameter 1116 for the last remaining ephemeral message 1102 within the ephemeral message group 1104 will determine when the ephemeral message group 1104 itself expires. In this case, a new ephemeral message 1102, added to the ephemeral message group 1104, with a new group participation parameter 1116, effectively extends the life of an ephemeral message group 1104 to equal the value of the group participation parameter 1116.

[0186] Responsive to the messaging system 1112 determining that an ephemeral message group 1104 has expired (e.g., is no longer accessible), the messaging system 1112 communicates with the interactive platform 100 (and, for example, specifically the interaction client 104) to cause an indicium (e.g., an icon) associated with the relevant ephemeral message group 1104 to no longer be displayed within a user interface of the interaction client 104. Similarly, when the messaging system 1112 determines that the message duration parameter 1106 for a particular ephemeral message 1102 has expired, the messaging system 1112 causes the interaction client 104 to no longer display an indicium (e.g., an icon or textual identification) associated with the ephemeral message 1102.

SYSTEM ARCHITECTURE

[0187] FIG. 12 is a block diagram illustrating further details regarding the interactive platform 100, according to some examples. Specifically, the interactive platform 100 is shown to include the interaction client 104 and the Interactive servers 124. The interactive platform 100 embodies multiple subsystems, which are supported on the client-side by the interaction client 104 and on the server-side by the Interactive servers 124. Example subsystems are discussed below.

[0188] An image processing system 1202 provides various functions that enable a user to capture and augment (e.g., augment or otherwise modify or edit) media content associated with a message.

[0189] A camera system 1204 includes control software (e.g., in a camera application) that interacts with and controls hardware camera hardware (e.g., directly or via operating system controls) of the user system 102 to modify and augment real-time images captured and displayed via the interaction client 104.

[0190] The augmentation system 1206 provides functions related to the generation and publishing of augmentations (e.g., media overlays) for images captured in real-time by cameras of the user system 102 or retrieved from memory of the user system 102. For example, the augmentation system 1206 operatively selects, presents, and displays media overlays (e.g., an image filter or an image lens) to the interaction client 104 for the augmentation of real-time images received via the camera system 1204 or stored images retrieved from memory of a user system 102. These augmentations are selected by the augmentation system 1206 and presented to a user of an interaction client 104, based on a number of inputs and data, such as, but not limited to, geolocation of the user system 102, and interactive platform information of the user of the user system 102, and the like.

[0191] An augmentation may include audio and visual content and visual effects. Examples of audio and visual content include pictures, texts, logos, animations, and sound effects. An example of a visual effect includes color overlaying. The audio and visual content or the visual effects can be applied to a media content item (e.g., a photo or video) at user system 102 for communication in a message, or applied to video content, such as a video content stream or feed

transmitted from an interaction client 104. As such, the image processing system 1202 may interact with, and support, the various subsystems of the communication system 1208, such as the messaging system 1210 and the video communication system 1212.

[0192] A media overlay may include text or image data that can be overlaid on top of a photograph taken by the user system 102 or a video stream produced by the user system 102. In some examples, the media overlay may be a location overlay (e.g., Venice beach), a name of a live event, or a name of a merchant overlay (e.g., Beach Coffee House). In further examples, the image processing system 1202 uses the geolocation of the user system 102 to identify a media overlay that includes the name of a merchant at the geolocation of the user system 102. The media overlay may include other indicia associated with the merchant. The media overlays may be stored in the databases 128 and accessed through the database server 126.

[0193] The image processing system 1202 provides a user-based publication platform that enables users to select a geolocation on a map and upload content associated with the selected geolocation. The user may also specify circumstances under which a particular media overlay should be offered to other users. The image processing system 1202 generates a media overlay that includes the uploaded content and associates the uploaded content with the selected geolocation.

[0194] The augmentation creation system 1214 supports augmented reality developer platforms and includes an application for content creators (e.g., artists and developers) to create and publish augmentations (e.g., augmented reality experiences) of the interaction client 104. The augmentation creation system 1214 provides a library of built-in features and tools to content creators including, for example custom shaders, tracking technology, and templates.

[0195] In some examples, the augmentation creation system 1214 provides a merchant-based publication platform that enables merchants to select a particular augmentation associated with a geolocation via a bidding process. For example, the augmentation creation system 1214 associates a media overlay of the highest bidding merchant with a corresponding geolocation for a predefined amount of time.

[0196] A communication system 1208 is responsible for enabling and processing multiple forms of communication and interaction within the interactive platform 100 and includes a messaging system 1210, a personal AI system 1230, an audio communication system 1216, and a video communication system 1212. The messaging system 1210 is responsible for enforcing the temporary or time-limited access to content by the interaction clients 104. The messaging system 1210 incorporates multiple timers within an ephemeral timer system (not shown) that, based on duration and display parameters associated with a message or collection of messages (e.g., a story), selectively enable access (e.g., for presentation and display) to messages and associated content via the interaction client 104. Further details regarding the operation of the ephemeral timer system are provided below. The audio communication system 1216 enables and supports audio communications (e.g., real-time audio chat) between multiple interaction clients 104. Similarly, the video communication system 1212 enables and supports video communications (e.g., real-time video chat) between multiple interaction clients 104. The personal AI system 1230 is responsible for generating responses to prompts received from a user and communicating a response to the prompt.

[0197] A user management system 1218 is operationally responsible for the management of user data and profiles, and maintains interactive platform information regarding relationships between users of the interactive platform 100.

[0198] A collection management system 1220 is operationally responsible for managing sets or collections of media (e.g., collections of text, image video, and audio data). A collection of content (e.g., messages, including images, video, text, and audio) may be organized into an “event gallery” or an “event story.” Such a collection may be made available for a specified time period, such as the duration of an event to which the content relates. For example, content relating to a music concert may be made available as a “story” for the duration of that music concert. The collection management system 1220 may also be responsible for publishing an icon that provides notification of a particular collection to the user interface of the interaction client 104. The collection management system 1220 includes a curation function that allows a collection manager to manage and curate a particular collection of content. For example, the curation interface enables an event organizer to curate a collection of content relating to a specific event (e.g.,

delete inappropriate content or redundant messages). Additionally, the collection management system 1220 employs machine vision (or image recognition technology) and content rules to curate a content collection automatically. In certain examples, compensation may be paid to a user to include user-generated content into a collection. In such cases, the collection management system 1220 operates to automatically make payments to such users to use their content.

[0199] A map system 1222 provides various geographic location functions and supports the presentation of map-based media content and messages by the interaction client 104. For example, the map system 1222 enables the display of user icons or avatars (e.g., stored in profile data 902) on a map to indicate a current or past location of "friends" of a user, as well as media content (e.g., collections of messages including photographs and videos) generated by such friends, within the context of a map. For example, a message posted by a user to the interactive platform 100 from a specific geographic location may be displayed within the context of a map at that particular location to "friends" of a specific user on a map interface of the interaction client 104. A user can furthermore share his or her location and status information (e.g., using an appropriate status avatar) with other users of the interactive platform 100 via the interaction client 104, with this location and status information being similarly displayed within the context of a map interface of the interaction client 104 to selected users.

[0200] A game system 1224 provides various gaming functions within the context of the interaction client 104. The interaction client 104 provides a game interface providing a list of available games that can be launched by a user within the context of the interaction client 104 and played with other users of the interactive platform 100. The interactive platform 100 further enables a particular user to invite other users to participate in the play of a specific game by issuing invitations to such other users from the interaction client 104. The interaction client 104 also supports audio, video, and text messaging (e.g., chats) within the context of gameplay, provides a leaderboard for the games, and also supports the provision of in-game rewards (e.g., coins and items).

[0201] An external resource system 1226 provides an interface for the interaction client 104 to communicate with remote servers (e.g., third-party servers 112) to launch or access external resources, i.e., applications or applets.

Each third-party server 112 hosts, for example, a markup language (e.g., HTML5) based application or a small-scale version of an application (e.g., game, utility, payment, or ride-sharing application). The interaction client 104 may launch a web-based resource (e.g., application) by accessing the HTML5 file from the third-party servers 112 associated with the web-based resource. Applications hosted by third-party servers 112 are programmed in JavaScript leveraging a Software Development Kit (SDK) provided by the Interactive servers 124. The SDK includes APIs with functions that can be called or invoked by the web-based application. The Interactive servers 124 host a JavaScript library that provides a given external resource access to specific user data of the interaction client 104. HTML5 is an example of technology for programming games, but applications and resources programmed based on other technologies can be used.

[0202] To integrate the functions of the SDK into the web-based resource, the SDK is downloaded by the third-party server 112 from the Interactive servers 124 or is otherwise received by the third-party server 112. Once downloaded or received, the SDK is included as part of the application code of a web-based external resource. The code of the web-based resource can then call or invoke certain functions of the SDK to integrate features of the interaction client 104 into the web-based resource.

[0203] The SDK stored on the interactive server system 110 effectively provides the bridge between an external resource (e.g., applications 106 or applets) and the interaction client 104. This gives the user a seamless experience of communicating with other users on the interaction client 104 while also preserving the look and feel of the interaction client 104. To bridge communications between an external resource and an interaction client 104, the SDK facilitates communication between third-party servers 112 and the interaction client 104. A WebViewJavaScriptBridge running on a user system 102 establishes two one-way communication channels between an external resource and the interaction client 104. Messages are sent between the external resource and the interaction client 104 via these communication channels asynchronously. Each SDK function invocation is sent as a message and callback. Each SDK function is implemented by constructing a unique callback identifier and sending a message with that callback identifier.

[0204] By using the SDK, not all information from the interaction client 104 is shared with third-party servers 112. The SDK limits which information is shared based on the needs of the external resource. Each third-party server 112 provides an HTML5 file corresponding to the web-based external resource to Interactive servers 124. The Interactive servers 124 can add a visual representation (such as a box art or other graphic) of the web-based external resource in the interaction client 104. Once the user selects the visual representation or instructs the interaction client 104 through a GUI of the interaction client 104 to access features of the web-based external resource, the interaction client 104 obtains the HTML5 file and instantiates the resources to access the features of the web-based external resource.

[0205] The interaction client 104 presents a graphical user interface (e.g., a landing page or title screen) for an external resource. During, before, or after presenting the landing page or title screen, the interaction client 104 determines whether the launched external resource has been previously authorized to access user data of the interaction client 104. In response to determining that the launched external resource has been previously authorized to access user data of the interaction client 104, the interaction client 104 presents another graphical user interface of the external resource that includes functions and features of the external resource. In response to determining that the launched external resource has not been previously authorized to access user data of the interaction client 104, after a threshold period of time (e.g., 3 seconds) of displaying the landing page or title screen of the external resource, the interaction client 104 slides up (e.g., animates a menu as surfacing from a bottom of the screen to a middle or other portion of the screen) a menu for authorizing the external resource to access the user data. The menu identifies the type of user data that the external resource will be authorized to use. In response to receiving a user selection of an accept option, the interaction client 104 adds the external resource to a list of authorized external resources and allows the external resource to access user data from the interaction client 104. The external resource is authorized by the interaction client 104 to access the user data under an OAuth 2 framework.

[0206] The interaction client 104 controls the type of user data that is shared with external resources based on the type of external resource being authorized. For example, external resources that include full-scale applications (e.g., an

application 106) are provided with access to a first type of user data (e.g., two-dimensional avatars of users with or without different avatar characteristics). As another example, external resources that include small-scale versions of applications (e.g., web-based versions of applications) are provided with access to a second type of user data (e.g., payment information, two-dimensional avatars of users, three-dimensional avatars of users, and avatars with various avatar characteristics). Avatar characteristics include different ways to customize a look and feel of an avatar, such as different poses, facial features, clothing, and so forth.

[0207] An advertisement system 1228 operationally enables the purchasing of advertisements by third parties for presentation to end-users via the interaction clients 104 and also handles the delivery and presentation of these advertisements.

SOFTWARE ARCHITECTURE

[0208] FIG. 13 is a block diagram 1300 illustrating a software architecture 1302, which can be installed on any one or more of the devices described herein.

The software architecture 1302 is supported by hardware such as a machine 1304 that includes processors 1306, memory 1308, and I/O components 1310. In this example, the software architecture 1302 can be conceptualized as a stack of layers, where each layer provides a particular functionality. The software architecture 1302 includes layers such as an operating system 1312, libraries 1314, frameworks 1316, and applications 1318. Operationally, the applications 1318 invoke API calls 1320 through the software stack and receive messages 1322 in response to the API calls 1320.

[0209] The operating system 1312 manages hardware resources and provides common services. The operating system 1312 includes, for example, a kernel 1324, services 1326, and drivers 1328. The kernel 1324 acts as an abstraction layer between the hardware and the other software layers. For example, the kernel 1324 provides memory management, processor management (e.g., scheduling), component management, networking, and security settings, among other functionalities. The services 1326 can provide other common services for the other software layers. The drivers 1328 are responsible for controlling or interfacing with the underlying hardware. For instance, the drivers 1328 can

include display drivers, camera drivers, BLUETOOTH® or BLUETOOTH® Low Energy drivers, flash memory drivers, serial communication drivers (e.g., USB drivers), WI-FI® drivers, audio drivers, power management drivers, and so forth.

[0210] The libraries 1314 provide a common low-level infrastructure used by the applications 1318. The libraries 1314 can include system libraries 1330 (e.g., C standard library) that provide functions such as memory allocation functions, string manipulation functions, mathematic functions, and the like. In addition, the libraries 1314 can include API libraries 1332 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 in two dimensions (2D) and three dimensions (3D) in a graphic content 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 1314 can also include a wide variety of other libraries 1334 to provide many other APIs to the applications 1318.

[0211] The frameworks 1316 provide a common high-level infrastructure that is used by the applications 1318. For example, the frameworks 1316 provide various graphical user interface (GUI) functions, high-level resource management, and high-level location services. The frameworks 1316 can provide a broad spectrum of other APIs that can be used by the applications 1318, some of which may be specific to a particular operating system or platform.

[0212] In an example, the applications 1318 may include a home application 1336, a contacts application 1338, a browser application 1340, a book reader application 1342, a location application 1344, a media application 1346, a messaging application 1348, a game application 1350, and a broad assortment of other applications such as a third-party application 1352. The applications 1318 are programs that execute functions defined in the programs. Various programming languages can be employed to create one or more of the applications 1318, structured in a variety of manners, such as object-oriented

programming languages (e.g., Objective-C, Java, or C++) or procedural programming languages (e.g., C or assembly language). In a specific example, the third-party application 1352 (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 1352 can invoke the API calls 1320 provided by the operating system 1312 to facilitate functionalities described herein.

[0213] Additional examples include:

Example 1 is a method comprising: receiving a user prompt from the user; determining a conversation history of one or more conversations between a user and a chatbot; generating an augmented prompt using the user prompt and the conversation history; communicating the augmented prompt to a generative AI model; receiving a response from the generative AI model to the augmented prompt; and providing the response to the user.

Example 2 is the subject matter of Example 1, wherein determining the conversation history comprises: retrieving, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot.

Example 3 is the subject matter of any of Examples 1-2, further comprising: generating one or more summarized memories using the conversation history.

Example 4 is the subject matter of any of Examples 1-3, further comprising: generating one or more moderated memories using the summarized memories; and storing the one or more moderated memories into a memories datastore.

Example 5 is the subject matter of any of Examples 1-4, further comprising, receiving a request from the user to delete the conversation history; and in response to receiving the request, deleting the conversation history and one or more memories stored in the memories datastore associated with the conversation history.

Example 6 is the subject matter of any of Examples 1-5, wherein generating the augmented prompt comprises: generating a current conversation context from a current conversation between the user and the chatbot; retrieving, from the memories datastore, one or more stored memories using the current conversation context; and generating the augmented prompt using the one or more stored memories.

Example 7 is the subject matter of any of Examples 1-6, wherein the one or more memories stored in the memories datastore are isolated for a first conversation group, and wherein the one or more memories are not used to generate the augmented prompt for a second conversation group.

Example 8 is at least one machine-readable medium including instructions that, when executed by processing circuitry, cause the processing circuitry to perform operations to implement any of Examples 1-7.

Example 9 is an apparatus comprising means to implement any of Examples 1-7.

Example 10 is a system to implement any of Examples 1-7.

CONCLUSION

[0214] Changes and modifications may be made to the disclosed examples without departing from the scope of the present disclosure. These and other changes or modifications are intended to be included within the scope of the present disclosure.

GLOSSARY

[0215] "Carrier signal" refers to any intangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine and includes digital or analog communications signals or other intangible media to facilitate communication of such instructions. Instructions may be transmitted or received over a network using a transmission medium via a network interface device.

[0216] "User device" refers to any machine that interfaces to a communications network to obtain resources from one or more server systems or other user devices. A user device may be, but is not limited to, a mobile phone, desktop

computer, laptop, portable digital assistants (PDAs), smartphones, tablets, ultrabooks, netbooks, laptops, multi-processor systems, microprocessor-based or programmable consumer electronics, game consoles, set-top boxes, or any other communication device that a user may use to access a network.

[0217] "Communication network" refers to one or more portions of a network that may 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), the Internet, a portion of the Internet, a portion of the Public Switched Telephone Network (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, a network or a portion of a network may include a wireless or cellular network, and the coupling may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or other types of cellular or wireless coupling. In this example, the coupling may 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.

[0218] "Component" refers to a device, physical entity, or logic having boundaries defined by function or subroutine calls, branch points, APIs, or other technologies that provide for the partitioning or modularization of particular processing or control functions. Components may be combined via their interfaces with other components to carry out a machine process. A component may be a packaged functional hardware unit designed for use with other components and a part of a program that usually performs a particular function of related functions. Components may constitute either software components (e.g., code embodied on a machine-readable medium) or hardware components. A

"hardware component" is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various examples, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware components of a computer system (e.g., a processor or a group of processors) may be configured by software (e.g., an application or application portion) as a hardware component that operates to perform certain operations as described herein. A hardware component may also be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware component may include dedicated circuitry or logic that is permanently configured to perform certain operations. A hardware component may be a special-purpose processor, such as a field-programmable gate array (FPGA) or an application-specific integrated circuit (ASIC). A hardware component may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware component may include software executed by a general-purpose processor or other programmable processors. Once configured by such software, hardware components become specific machines (or specific components of a machine) uniquely tailored to perform the configured functions and are no longer general-purpose processors. It will be appreciated that the decision to implement a hardware component mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software), may be driven by cost and time considerations. Accordingly, the phrase "hardware component" (or "hardware-implemented component") should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein. Considering examples in which hardware components are temporarily configured (e.g., programmed), each of the hardware components need not be configured or instantiated at any one instance in time. For example, where a hardware component comprises a general-purpose processor configured by software to become a special-purpose processor, the general-purpose processor may be configured as respectively different special-purpose processors (e.g., comprising different hardware components) at different times. Software accordingly

configures a particular processor or processors, for example, to constitute a particular hardware component at one instance of time and to constitute a different hardware component at a different instance of time. Hardware components can provide information to, and receive information from, other hardware components. Accordingly, the described hardware components may be regarded as being communicatively coupled. Where multiple hardware components exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) between or among two or more of the hardware components. In examples in which multiple hardware components are configured or instantiated at different times, communications between such hardware components may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware components have access. For example, one hardware component may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware component may then, at a later time, access the memory device to retrieve and process the stored output. Hardware components may also initiate communications with input or output devices, and can operate on a resource (e.g., a collection of information). The various operations of example methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented components that operate to perform one or more operations or functions described herein. As used herein, "processor-implemented component" refers to a hardware component implemented using one or more processors. Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented components. Moreover, the one or more processors may also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service" (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines including processors), with these operations being

accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., an API). The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some examples, the processors or processor-implemented components may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other examples, the processors or processor-implemented components may be distributed across a number of geographic locations.

[0219] "Machine-readable storage medium" refers to both machine-storage media and transmission media. Thus, the terms include both storage devices/media and carrier waves/modulated data signals. The terms "computer-readable medium," "machine-readable medium" and "device-readable medium" mean the same thing and may be used interchangeably in this disclosure.

[0220] "Ephemeral message" refers to a message that is accessible for a time-limited duration. An ephemeral message may be a text, an image, a video and the like. The access time for the ephemeral message may be set by the message sender. Alternatively, the access time may be a default setting or a setting specified by the recipient. Regardless of the setting technique, the message is transitory.

[0221] "Machine storage medium" refers to a single or multiple storage devices and media (e.g., a centralized or distributed database, and associated caches and servers) that store executable instructions, routines and data. The term shall accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media, including memory internal or external to processors. Specific examples of machine-storage media, computer-storage media and device-storage media include non-volatile memory, including by way of example semiconductor memory devices, e.g., erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), FPGA, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The terms "machine-storage medium," "device-storage medium," "computer-storage medium" mean the same thing and may be used interchangeably in this disclosure. The terms "machine-storage media," "computer-storage media," and

"device-storage media" specifically exclude carrier waves, modulated data signals, and other such media, at least some of which are covered under the term "signal medium."

[0222] "Non-transitory machine-readable storage medium" refers to a tangible medium that is capable of storing, encoding, or carrying the instructions for execution by a machine.

[0223] "Signal medium" refers to any intangible medium that is capable of storing, encoding, or carrying the instructions for execution by a machine and includes digital or analog communications signals or other intangible media to facilitate communication of software or data. The term "signal medium" shall be taken to include any form of a modulated data signal, carrier wave, and so forth. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. The terms "transmission medium" and "signal medium" mean the same thing and may be used interchangeably in this disclosure.

[0224] A "chatbot", also known as an interactive software application or digital agent, may refer to a computer-implemented system designed to engage with users through conversational interfaces using pre-determined scripts and/or artificial intelligence (AI) methodologies including but not limited to natural language processing (NLP), natural language understanding (NLU), and machine learning (ML). A chatbot may operate by receiving user inputs in the form of text or voice, processing these inputs according to algorithms and logic rules, interpreting the user's intent, and generating appropriate responses. The responses can range from simple, fixed text-based replies to complex, contextually relevant, and dynamic conversation flows. A chatbot may interact through various platforms such as web interfaces, mobile applications, social media messaging platforms, email, SMS, or other digital communication channels. Furthermore, a chatbot may be designed to perform specific tasks like answering queries, providing recommendations, executing transactions, or directing users to relevant resources based on user interaction. It may also employ user-specific personalization by incorporating user context, past interactions, and preferences, which may be stored in an associated data repository, into its decision-making processes. While a chatbot may function in standalone fashion, it can also be

integrated with other software systems, databases, and services, including but not limited to customer relationship management (CRM) systems, content management systems (CMS), data management platforms (DMP), and other backend systems or third-party services, to access and exchange information, thereby extending its capabilities. This system may include one or more processors configured to execute instructions, stored in a memory component, for performing the various operations associated with receiving, processing, and responding to user inputs in a conversational manner.

[0225] A "memory" may refer, in the context of a conversational history on an interaction platform, to a synthesized representation of a user's past interactions, experiences, behaviors, and preferences as discerned from their communication record within the interactive platform. A memory representation may be constructed from the aggregation and analysis of conversational data points, including text inputs, response choices, dialogue duration, timestamp, frequency, context, sentiment, and any other accessible user-specific or interaction-related information. This data might be culled from various interaction channels such as instant messaging, voice or video calls, emails, social media interactions, or other digital communication interfaces offered by the platform. Inferred from these data points, a memory representation may encapsulate a user's recurring themes, topics of interest, sentiment trends, preferred interaction styles, and personal attributes. This information is intended to enrich the platform's understanding of the user, enabling more tailored, contextual, and effective interactions.

[0226] In this disclosure and appended claims, the terms "a" or "an" are used, as is common in patent documents, to include one or more than one, independent of any other instances or usages of "at least one" or "one or more." In this disclosure and appended claims, the term "or" is used to refer to a nonexclusive or, such that "A or B" includes "A but not B," "B but not A," and "A and B," unless otherwise indicated. In the appended claims, the terms "including" and "in which" are used as the plain-English equivalents of the respective terms "including" and "wherein." Also, in the appended claims, the terms "including" and "including" are open-ended; that is, a system, device, article, or process that includes elements in addition to those listed after such a term in a claim is still deemed to fall within the scope of that claim.

What is claimed is:

1. A method comprising:

receiving, by one or more processors, a user prompt from a user;
determining, by the one or more processors, a conversation history of one or more conversations between the user and a chatbot;
generating, by the one or more processors, an augmented prompt using the user prompt and the conversation history;
communicating, by the one or more processors, the augmented prompt to a generative AI model;
receiving, by the one or more processors, a response from the generative AI model to the augmented prompt; and
providing, by the one or more processors, the response to the user.

2. The method of claim 1, wherein determining the conversation history comprises:

retrieving, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot.

3. The method of claim 1, further comprising:

generating one or more summarized memories using the conversation history.

4. The method of claim 3, further comprising:

generating one or more moderated memories using the summarized memories; and
storing the one or more moderated memories into a memories datastore.

5. The method of claim 4, further comprising,

receiving a request from the user to delete the conversation history; and
in response to receiving the request, deleting the conversation history and one or more memories stored in the memories datastore associated with the conversation history.

6. The method of claim 4, wherein generating the augmented prompt comprises:

generating a current conversation context from a current conversation between the user and the chatbot;
retrieving, from the memories datastore, one or more stored memories using the current conversation context; and
generating the augmented prompt using the one or more stored memories.

7. The method of claim 6, wherein the one or more memories stored in the memories datastore are isolated for a first conversation group, and
wherein the one or more memories are not used to generate the augmented prompt for a second conversation group.

8. A machine, comprising:

one or more processors; and
one or more memories storing instructions that, when executed by the one or more processors, cause the machine to perform operations comprising:
receiving, by one or more processors, a user prompt from a user;
determining, by the one or more processors, a conversation history of one or more conversations between the user and a chatbot;
generating, by the one or more processors, an augmented prompt using the user prompt and the conversation history;
communicating, by the one or more processors, the augmented prompt to a generative AI model;
receiving, by the one or more processors, a response from the generative AI model to the augmented prompt; and
providing, by the one or more processors, the response to the user.

9. The machine of claim 8, wherein determining the conversation history comprises:

retrieving, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot.

10. The machine of claim 8, wherein the operations further comprise:

generating one or more summarized memories using the conversation history.

11. The machine of claim 10, wherein the operations further comprise:
 - generating one or more moderated memories using the summarized memories; and
 - storing the one or more moderated memories into a memories datastore.
12. The machine of claim 11, wherein the operations further comprise,
 - receiving a request from the user to delete the conversation history; and
 - in response to receiving the request, deleting the conversation history and one or more memories stored in the memories datastore associated with the conversation history.
13. The machine of claim 11, wherein the operations further comprise:
 - generating a current conversation context from a current conversation between the user and the chatbot;
 - retrieving, from the memories datastore, one or more stored memories using the current conversation context; and
 - generating the augmented prompt using the one or more stored memories.
14. The machine of claim 13, wherein the one or more memories stored in the memories datastore are isolated for a first conversation group, and
 - wherein the one or more memories are not used to generate the augmented prompt for a second conversation group.
15. A machine-storage medium storing machine-executable instructions that, when executed by a machine, cause the machine to perform operations comprising:
 - receiving a user prompt from a user;
 - determining a conversation history of one or more conversations between the user and a chatbot;
 - generating an augmented prompt using the user prompt and the conversation history;
 - communicating the augmented prompt to a generative AI model;
 - receiving a response from the generative AI model to the augmented prompt; and
 - providing the response to the user.

16. The machine-storage medium of claim 15, wherein determining the conversation history comprises:

retrieving, from a conversation history datastore, a conversation history of one or more conversations between a user and a chatbot.

17. The machine-storage medium of claim 15, wherein the operations further comprise:

generating one or more summarized memories using the conversation history.

18. The machine-storage medium of claim 17, wherein the operations further comprise:

generating one or more moderated memories using the summarized memories; and

storing the one or more moderated memories into a memories datastore.

19. The machine-storage medium of claim 18, wherein generating the augmented prompt comprises:

generating a current conversation context from a current conversation between the user and the chatbot;

retrieving, from the memories datastore, one or more stored memories using the current conversation context; and

generating the augmented prompt using the one or more stored memories.

20. The machine-storage medium of claim 19, wherein the one or more memories stored in the memories datastore are isolated for a first conversation group, and

wherein the one or more memories are not used to generate the augmented prompt for a second conversation group.

1/15

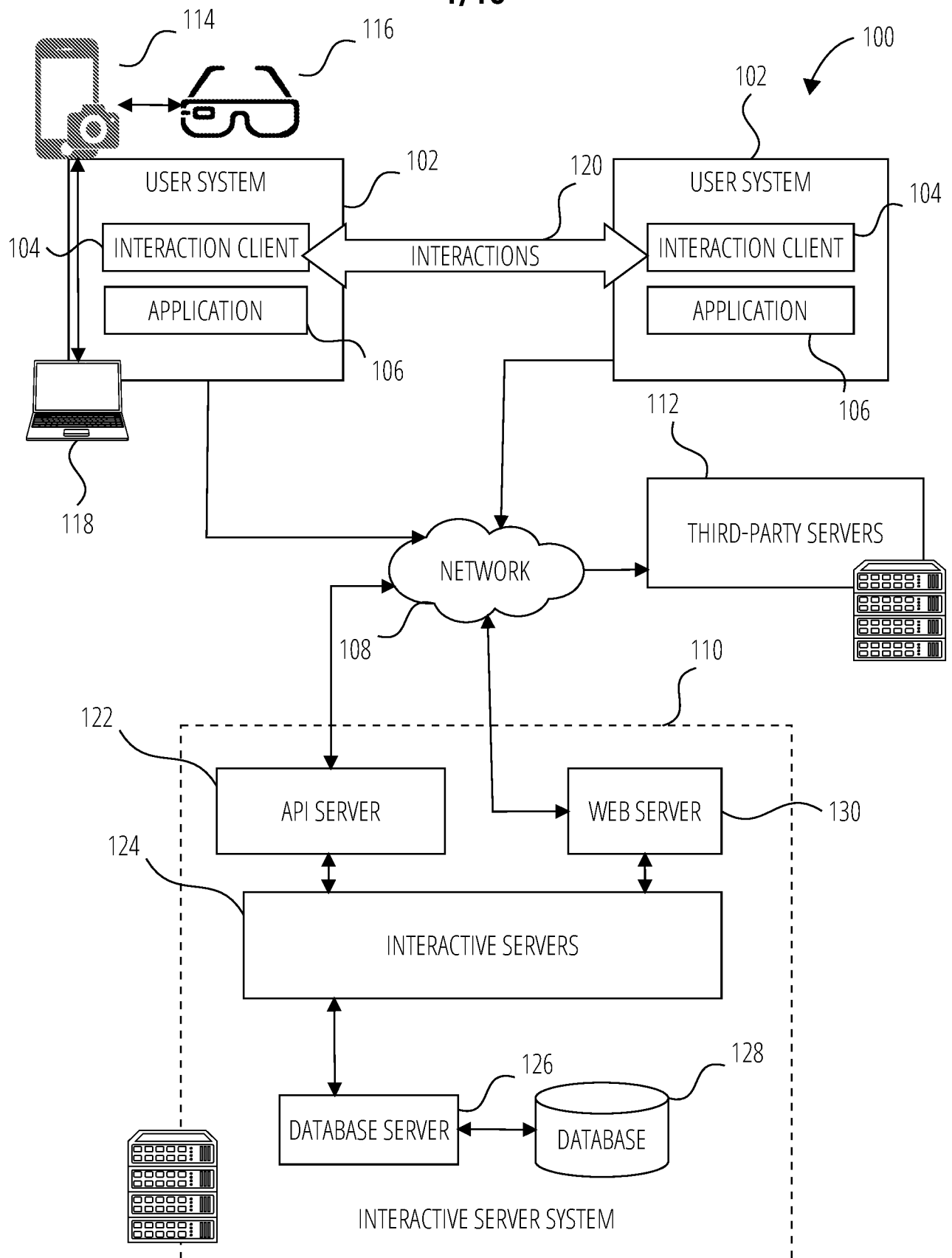


FIG. 1

2/15

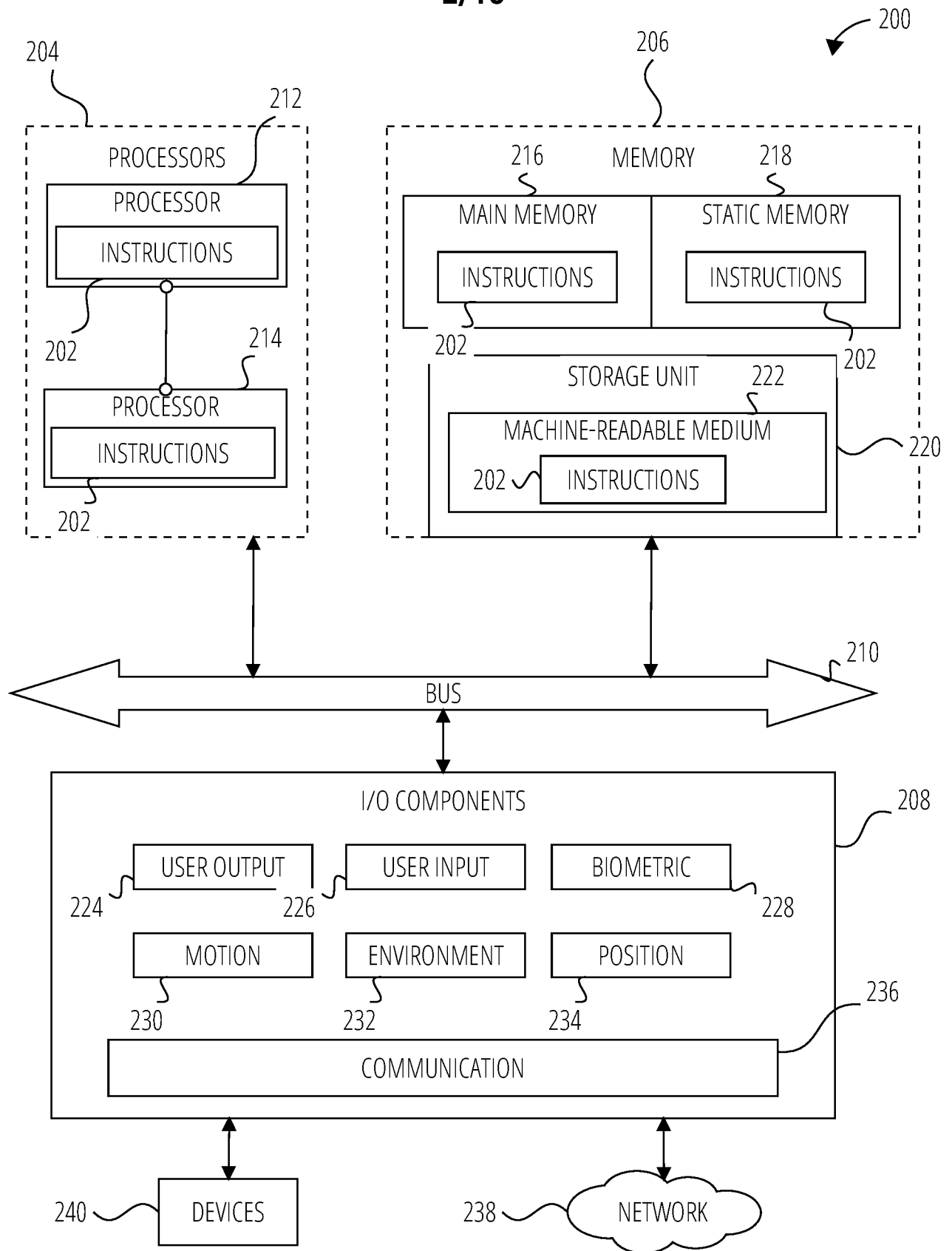


FIG. 2

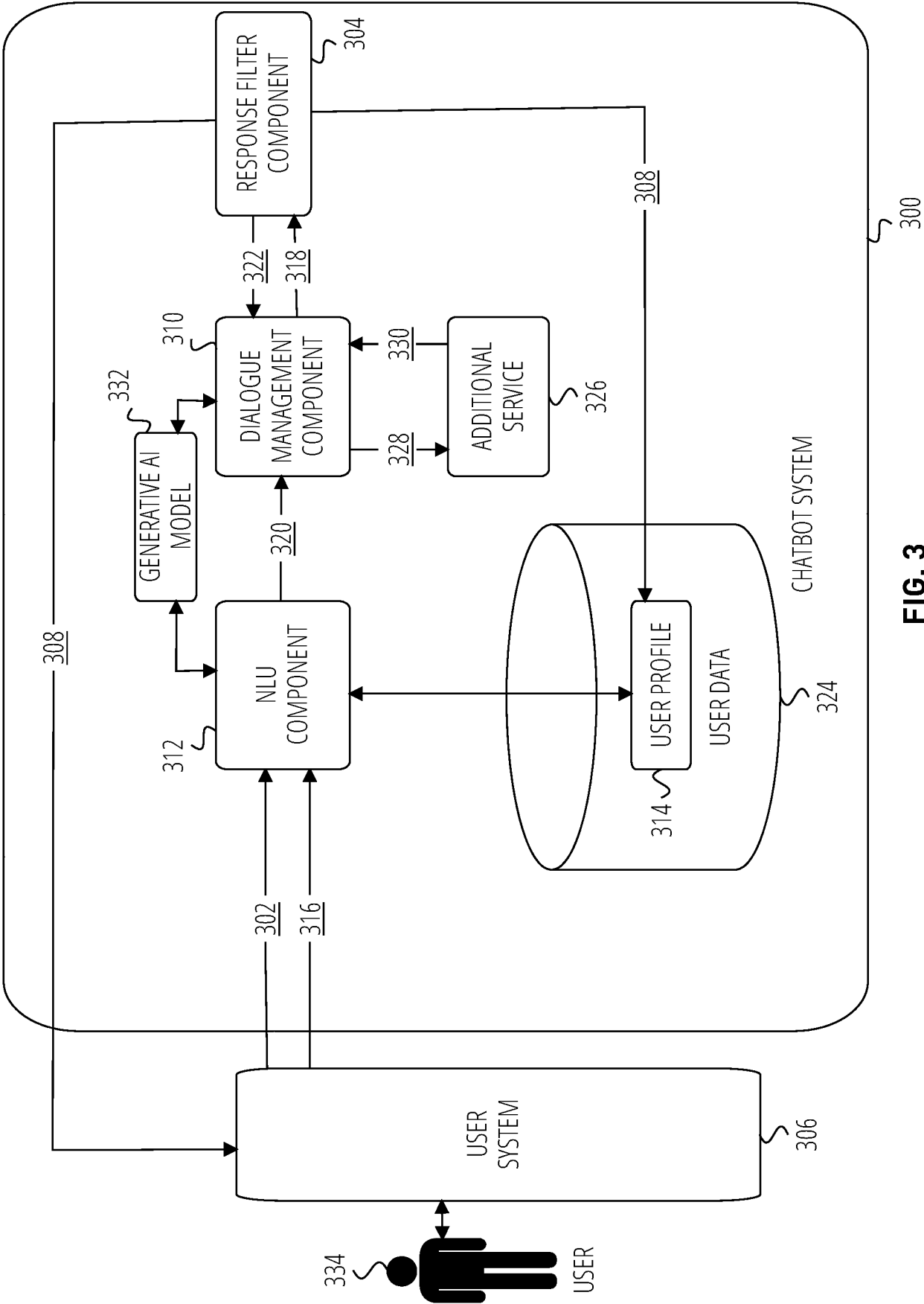


FIG. 3

4/15

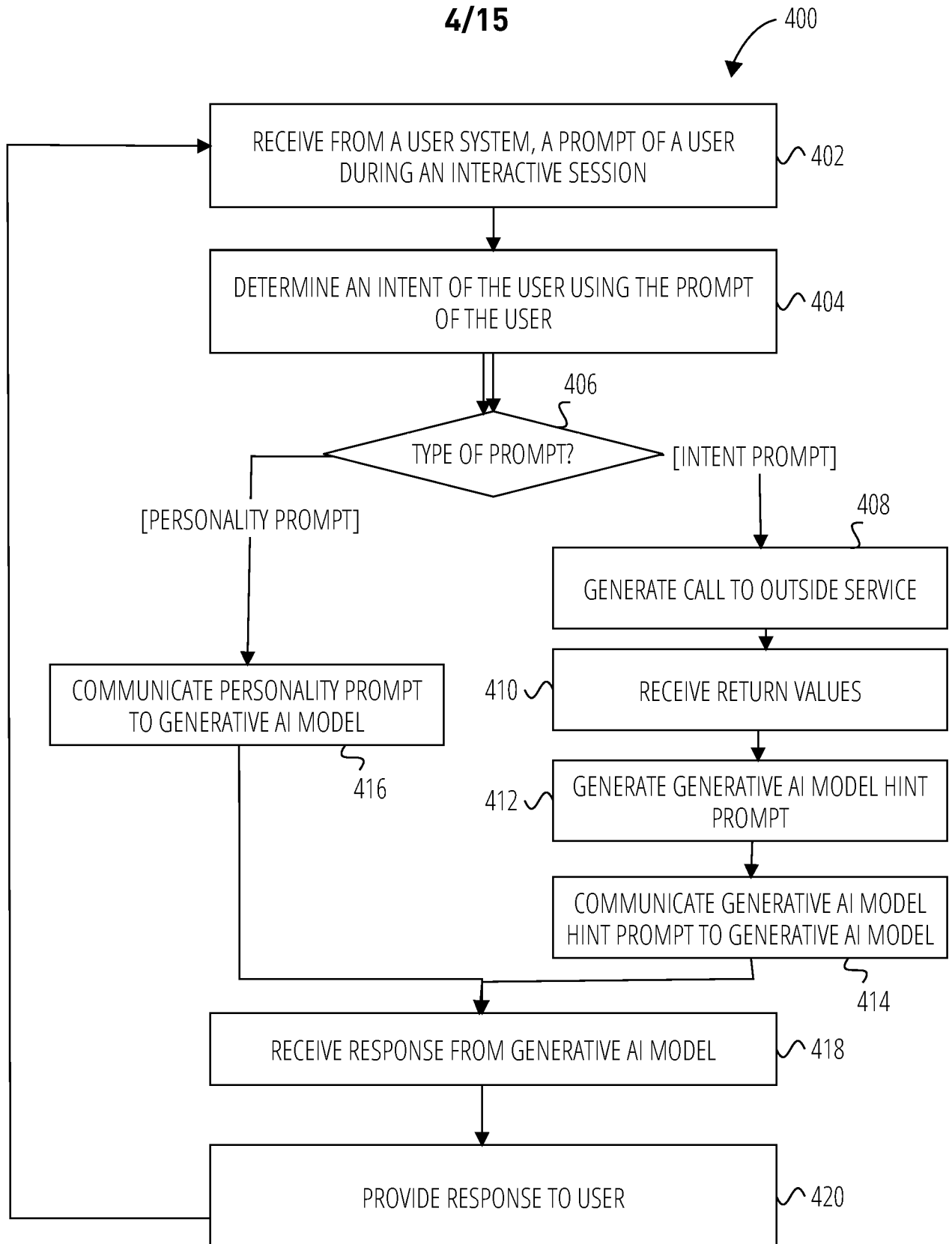


FIG. 4A

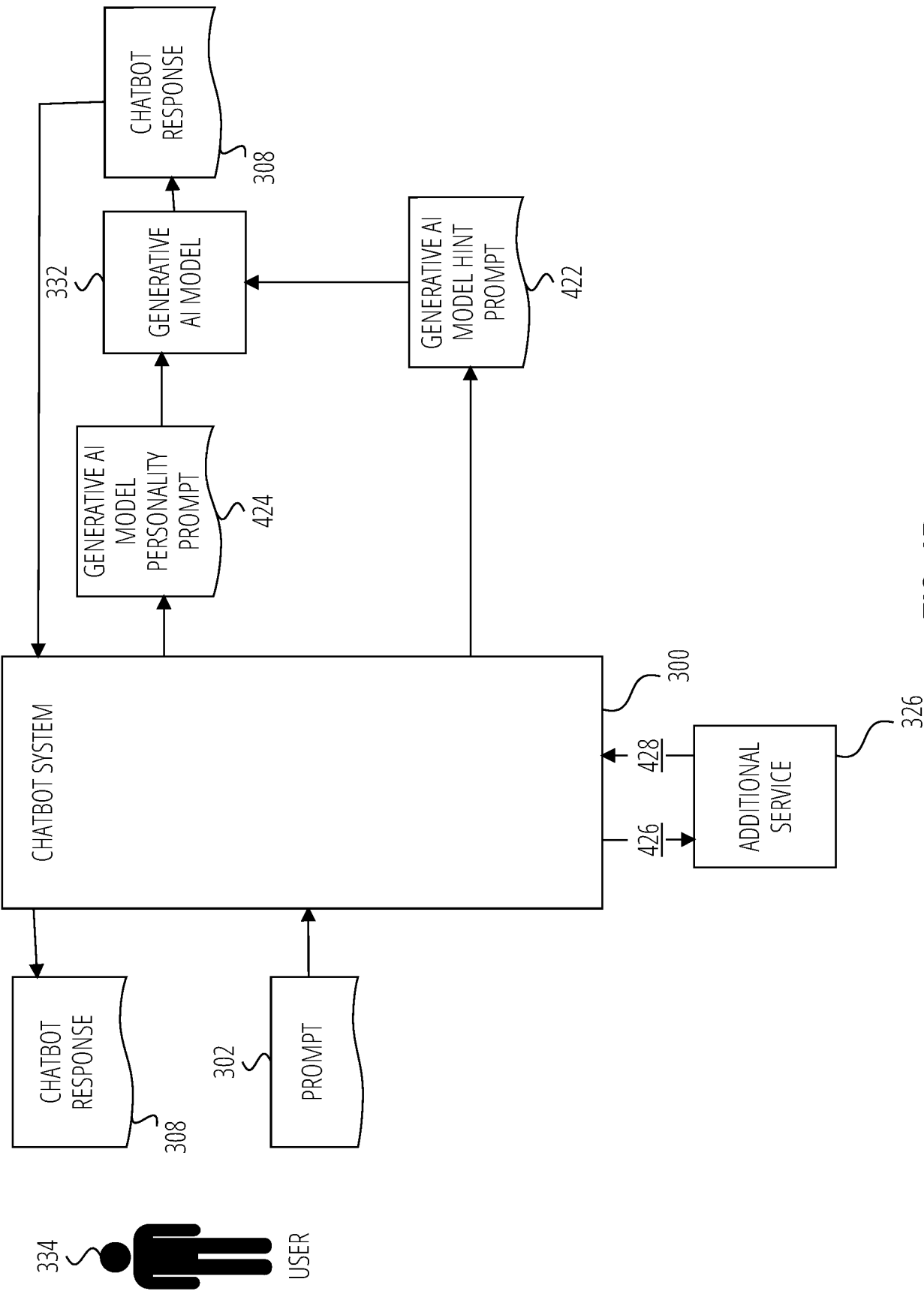


FIG. 4B

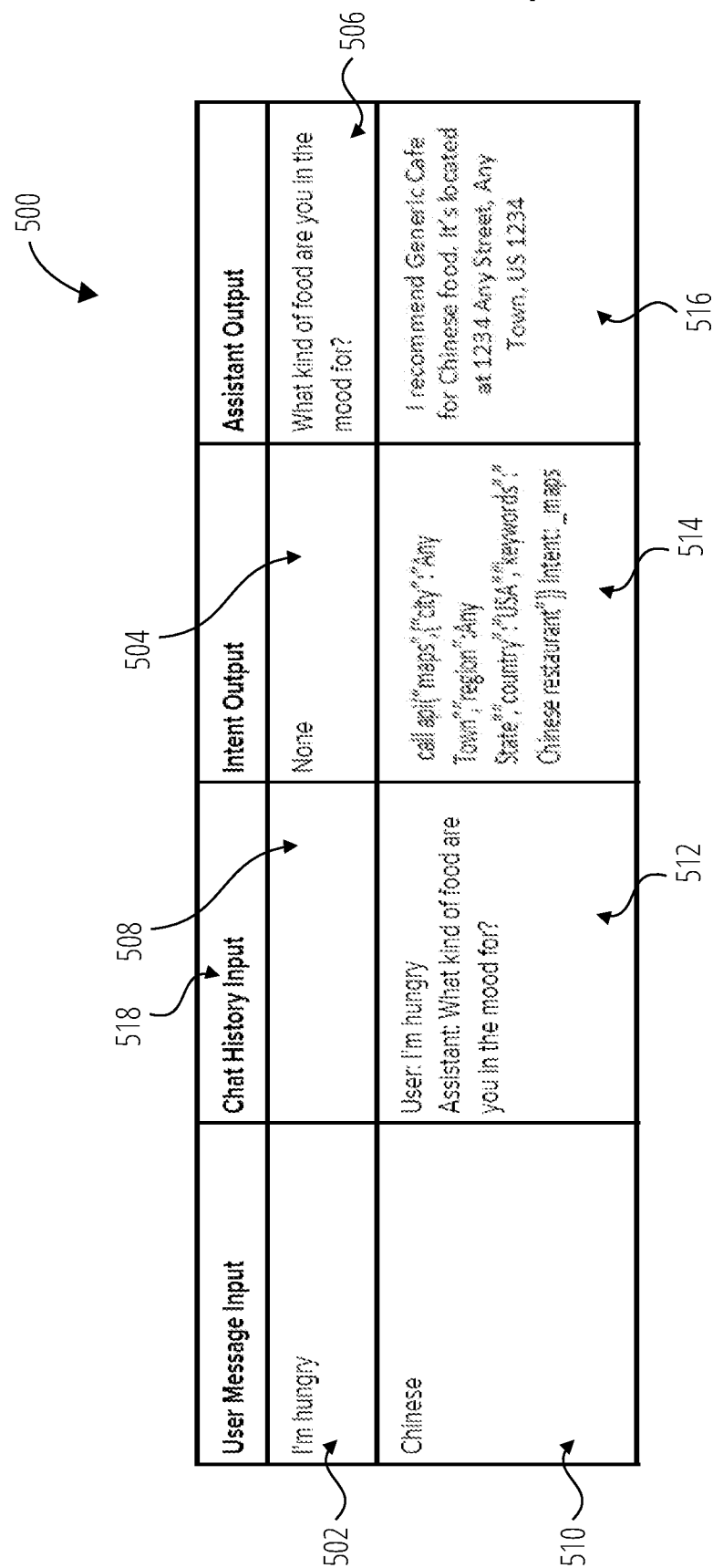


FIG. 5

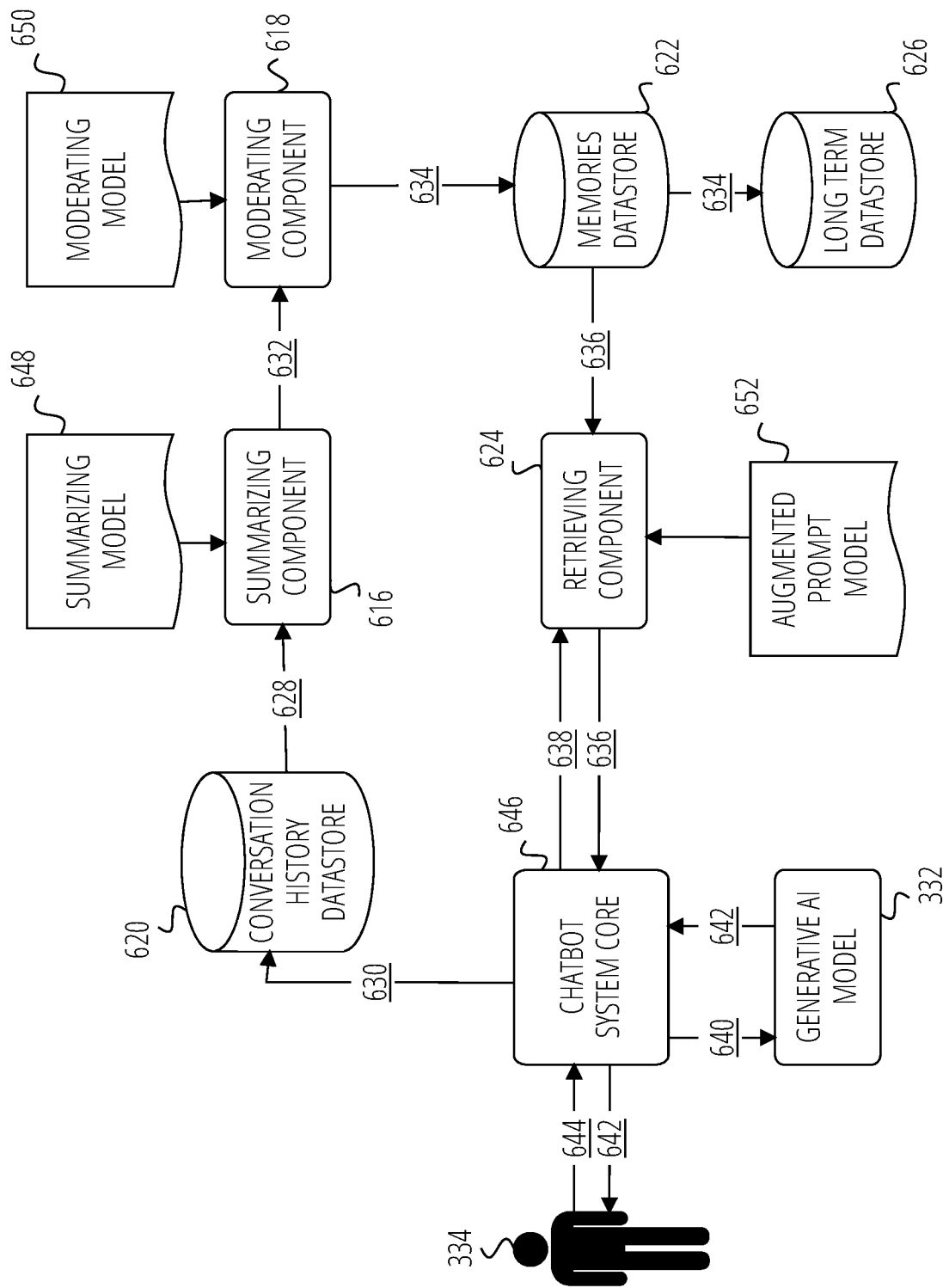


FIG. 6A

8/15

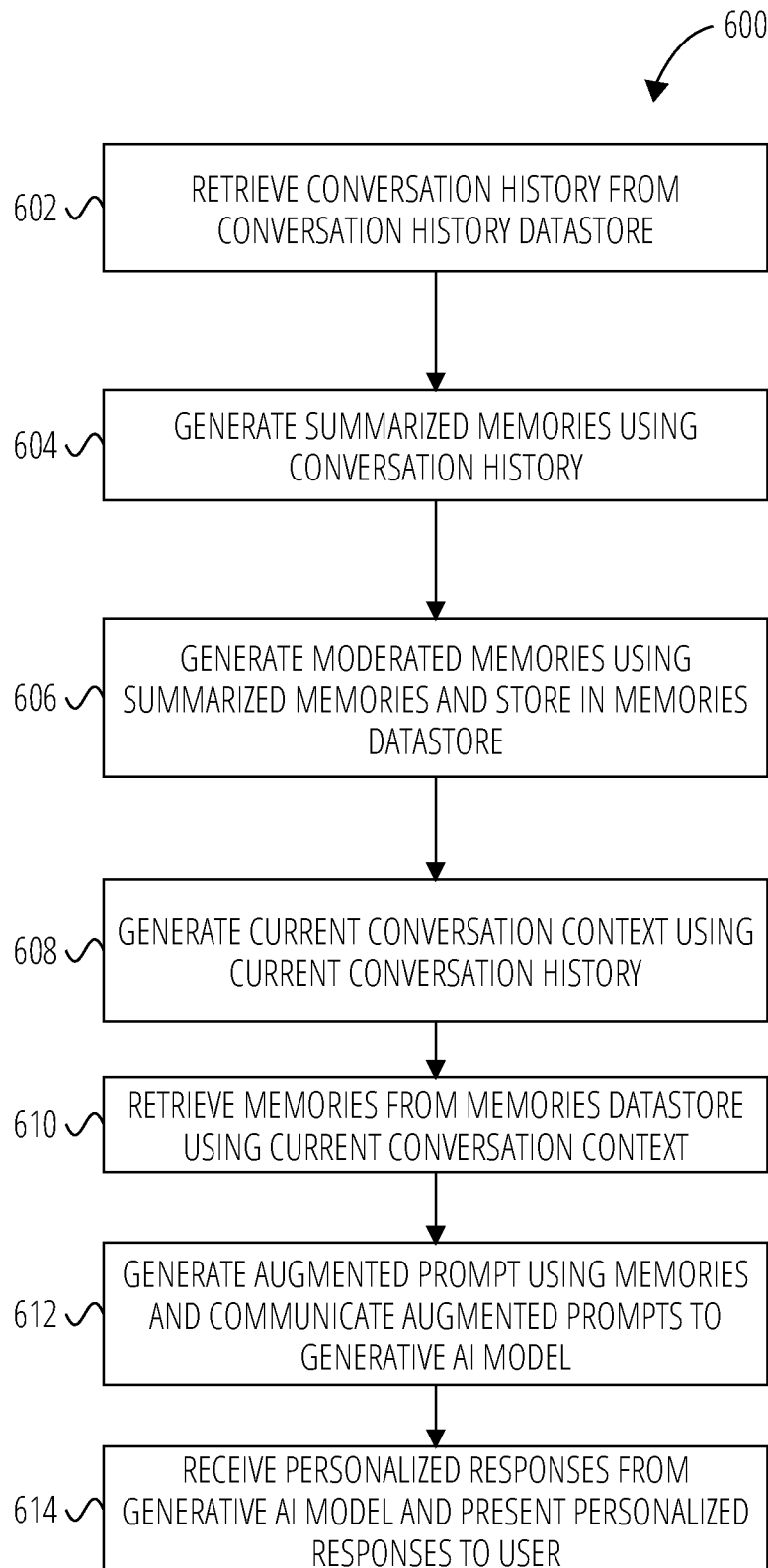


FIG. 6B

9/15

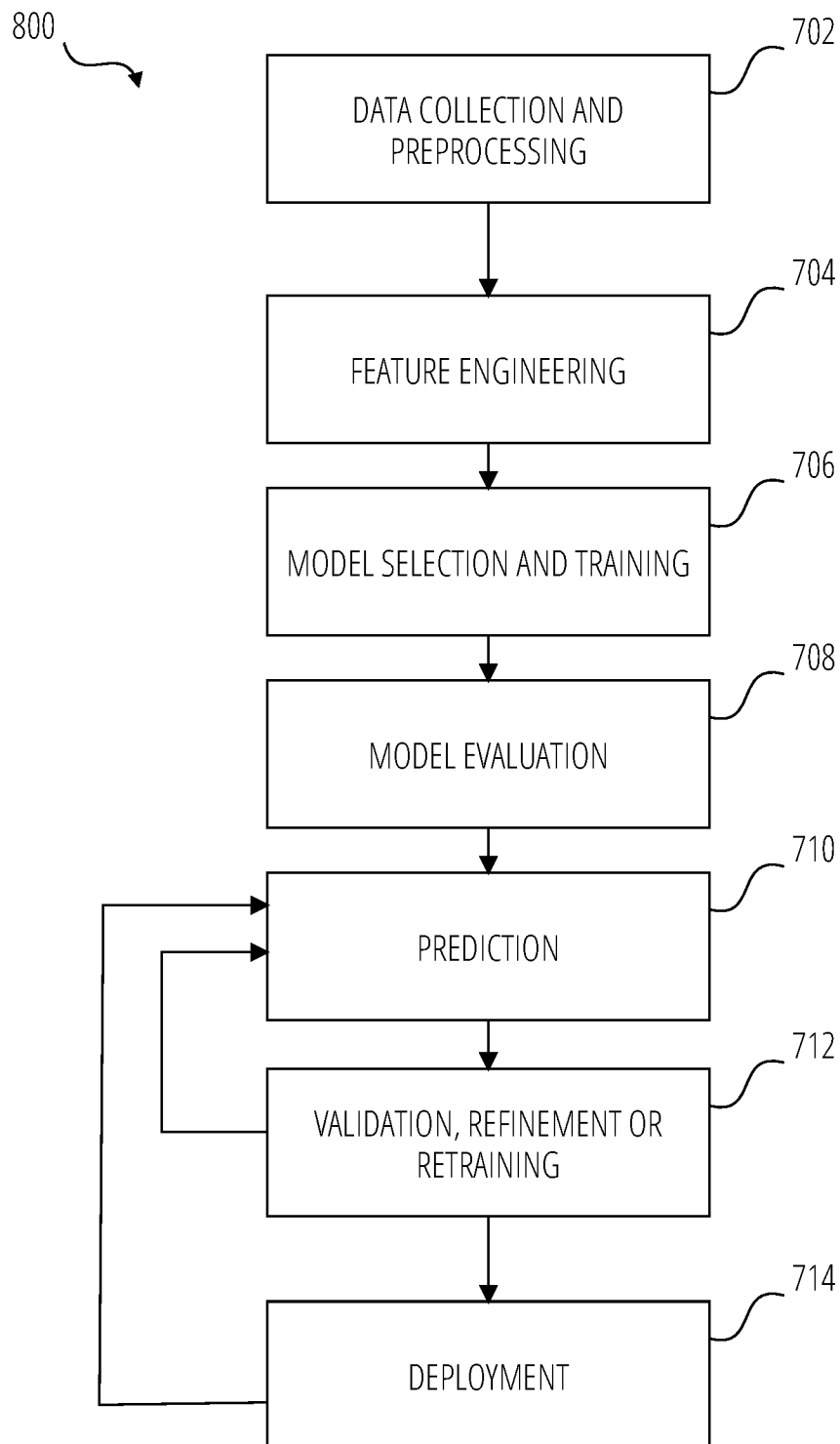


FIG. 7

10/15

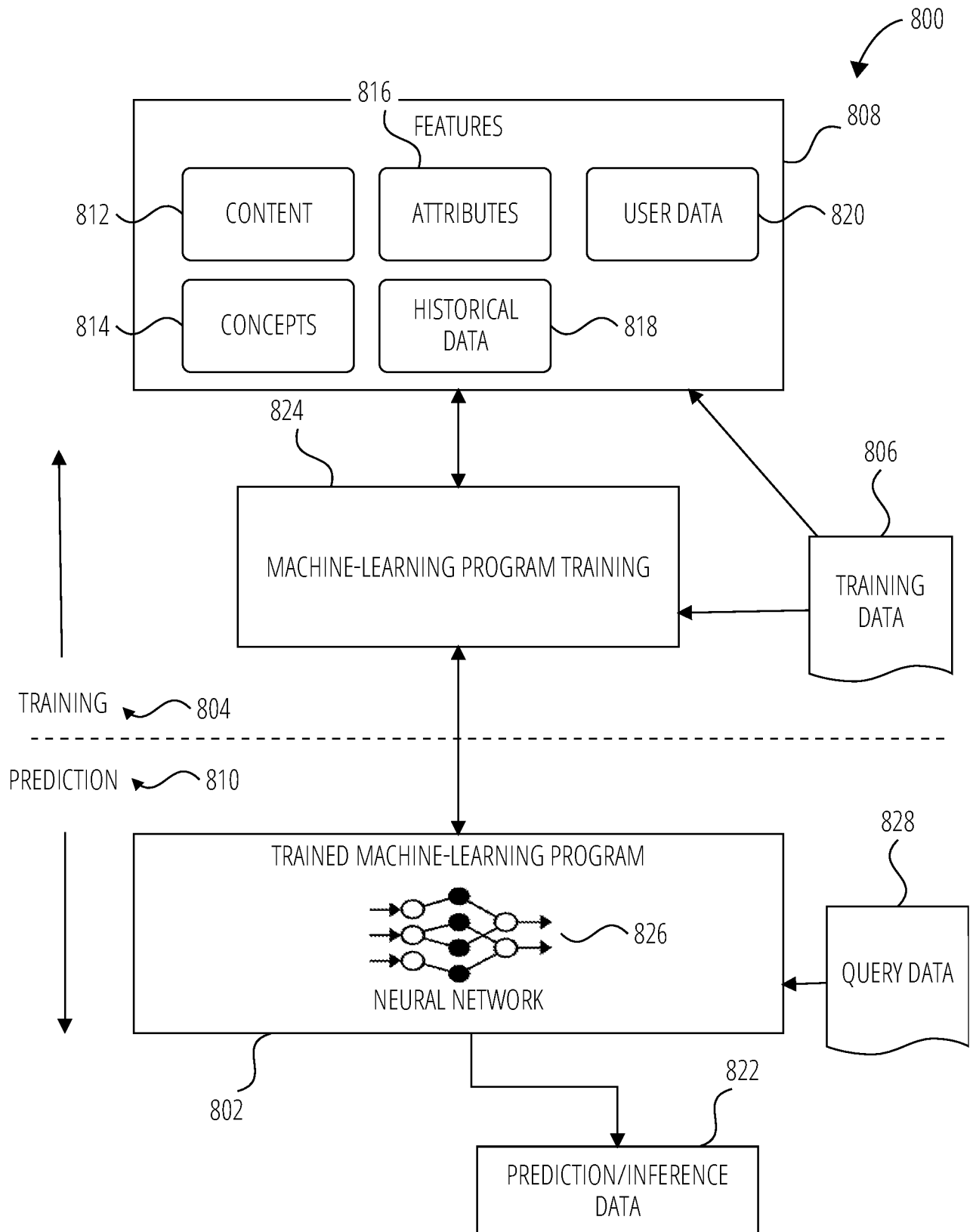


FIG. 8

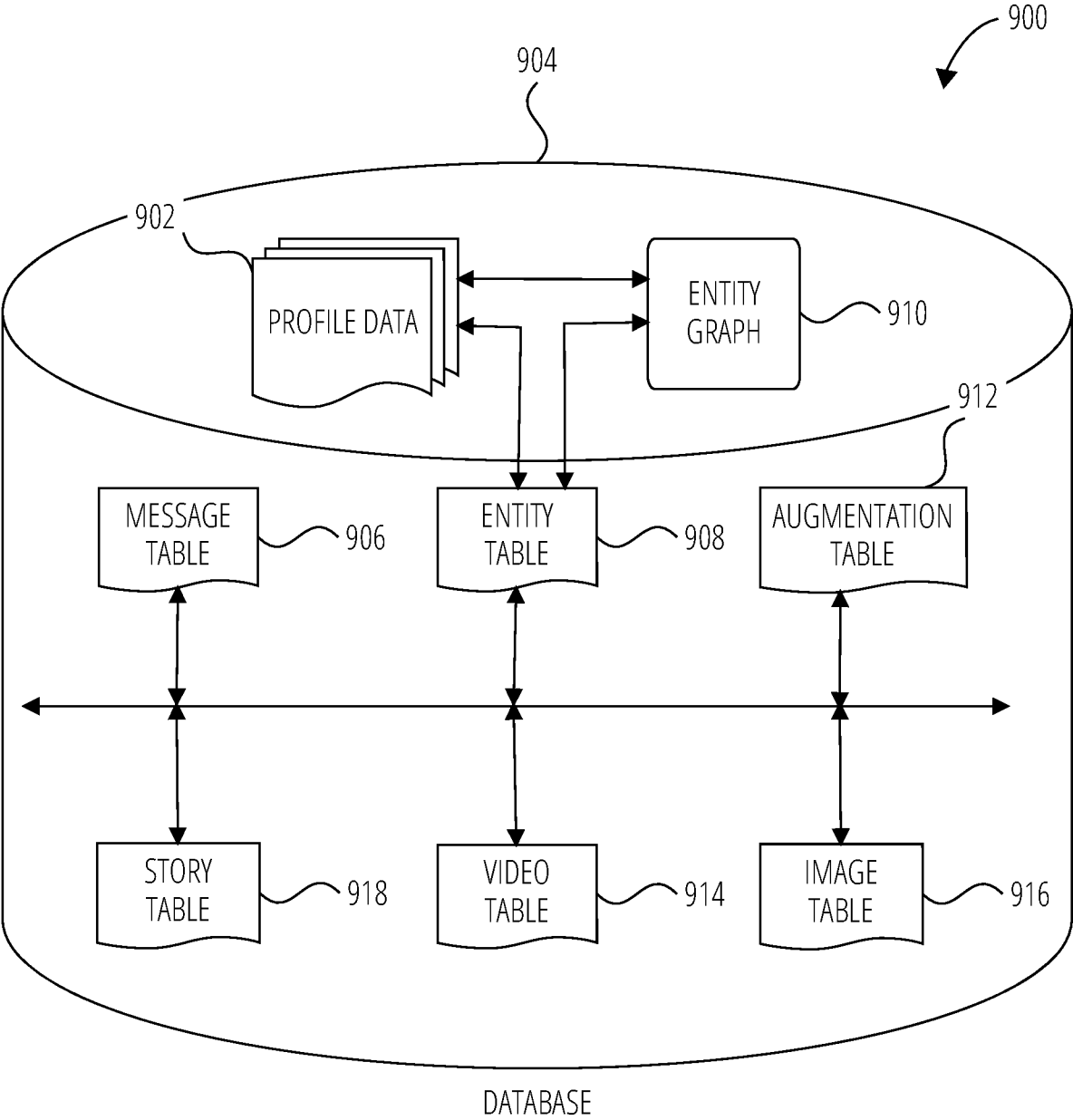


FIG. 9

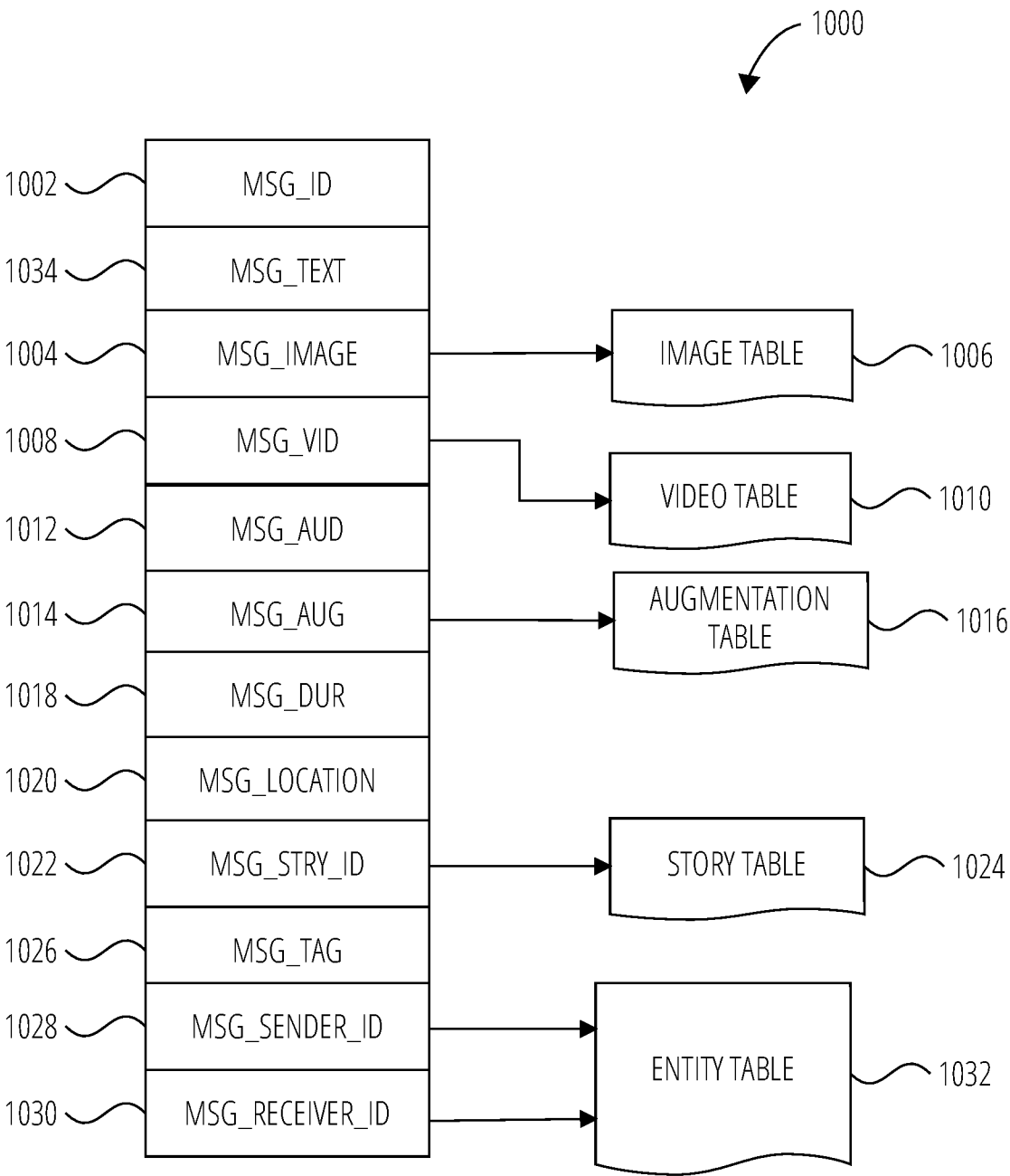


FIG. 10

13/15

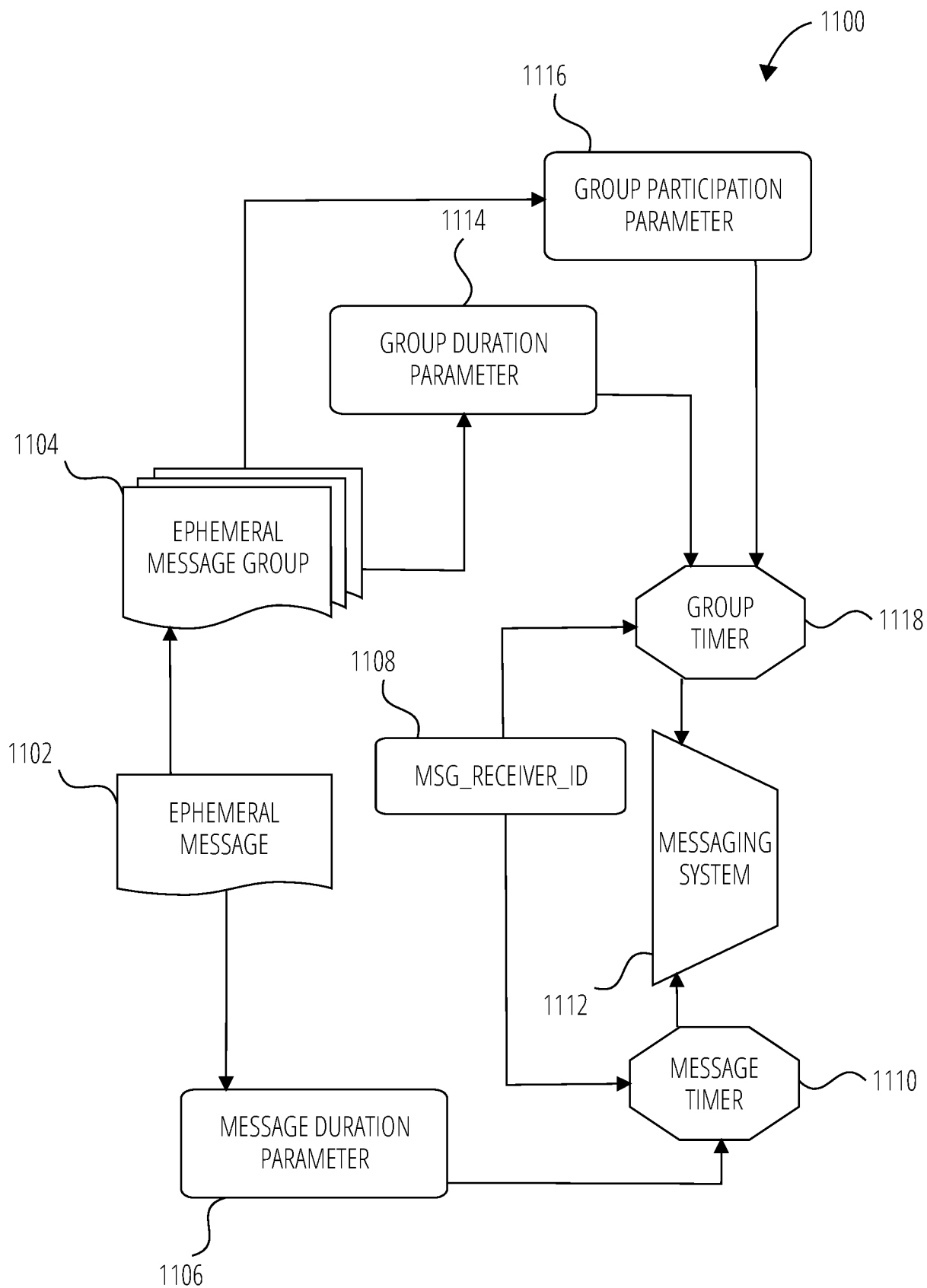


FIG. 11

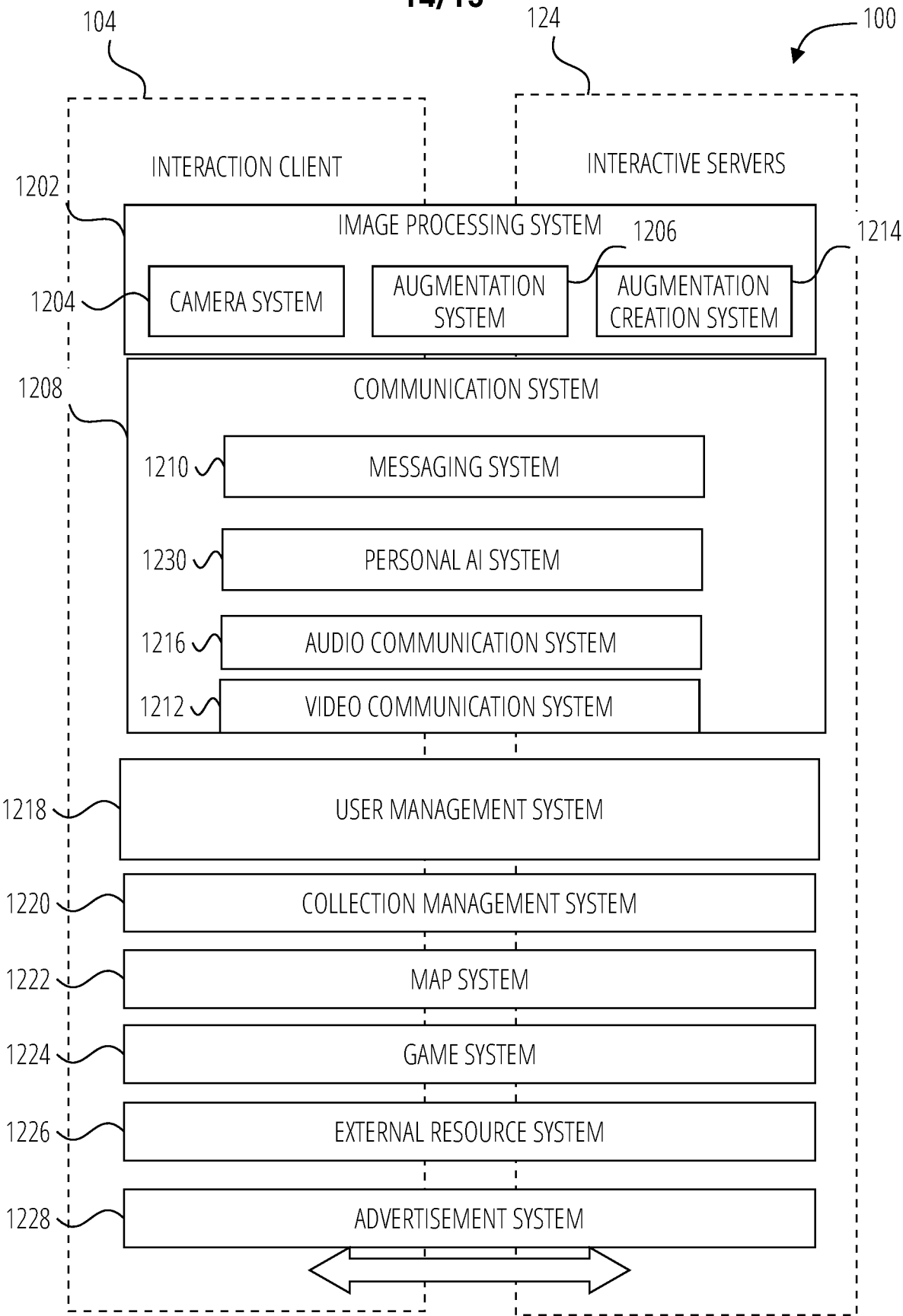


FIG. 12

15/15

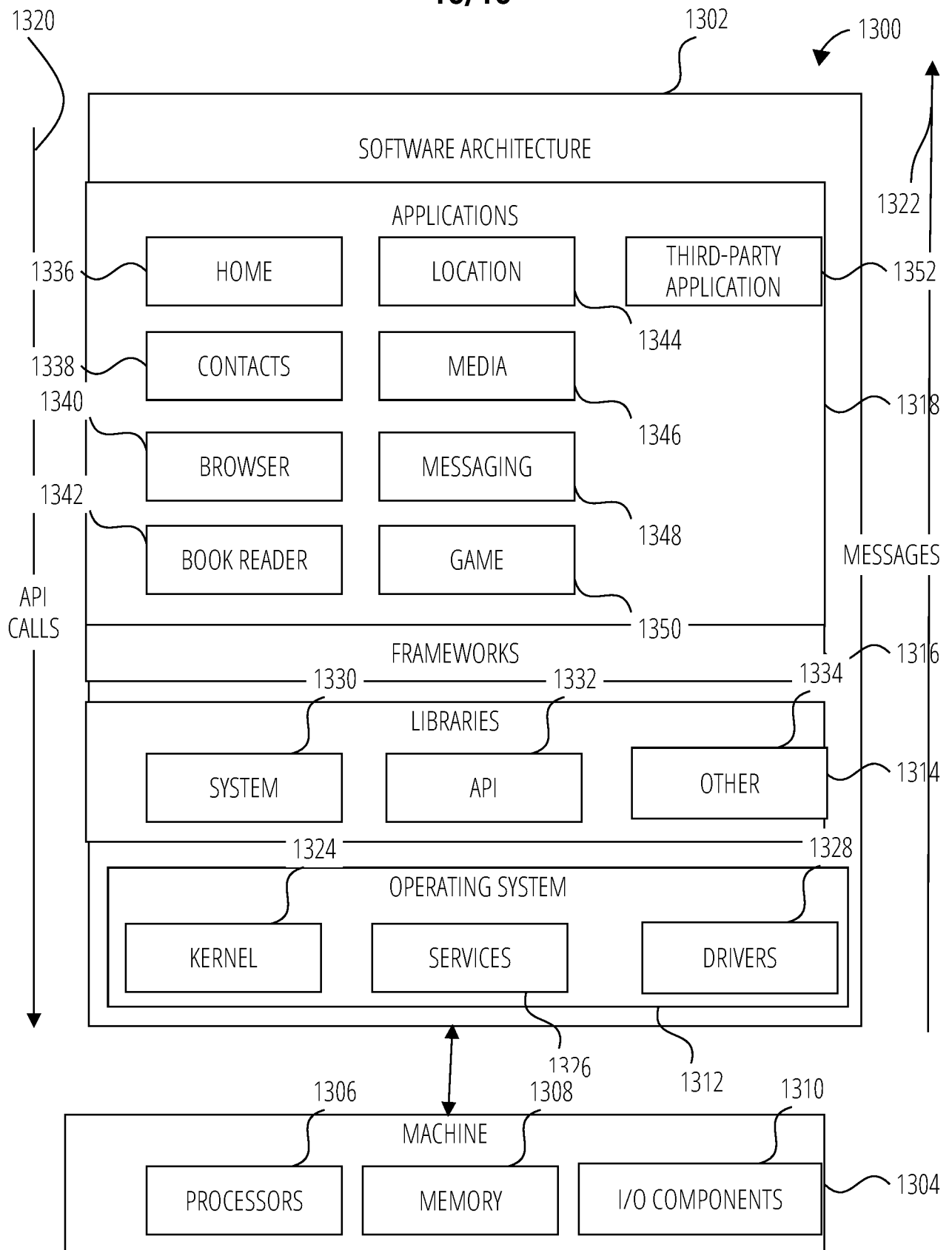


FIG. 13

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2024/032878

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F40/35 H04L51/02 H04L51/216
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F H04L G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2022/293107 A1 (LEAMAN IAN [US] ET AL) 15 September 2022 (2022-09-15) paragraph [0128] paragraph [0181] paragraph [0155] paragraph [0224] paragraph [0337] paragraph [0368] -----	1 - 20

☐

Further documents are listed in the continuation of Box C.

☒

See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

13 September 2024

Date of mailing of the international search report

26/09/2024

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Krawaritis, Achilles

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2024/032878

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2022293107 A1	15-09-2022	US 2022292423 A1	15-09-2022
		US 2022292465 A1	15-09-2022
		US 2022292525 A1	15-09-2022
		US 2022293107 A1	15-09-2022
