



- (51) **International Patent Classification:**
G10L 15/26 (2006.01)
- (21) **International Application Number:**
PCT/US2013/044834
- (22) **International Filing Date:**
7 June 2013 (07.06.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/657,723 8 June 2012 (08.06.2012) US
13/913,173 7 June 2013 (07.06.2013) US
- (71) **Applicant:** APPLE INC. [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).
- (72) **Inventor:** KEEN, Daniel; 1 Infinite Loop, Cupertino, CA 95014 (US).
- (74) **Agents:** BREGMAN, Dion, M. et al.; Morgan Lewis & Bockius LLP, 2 Palo Alto Square, 3000 El Camino Real, Suite 700, Palo Alto, CA 94306 (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, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, 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, 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:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

- (54) **Title:** SYSTEMS AND METHODS FOR RECOGNIZING TEXTUAL IDENTIFIERS WITHIN A PLURALITY OF WORDS

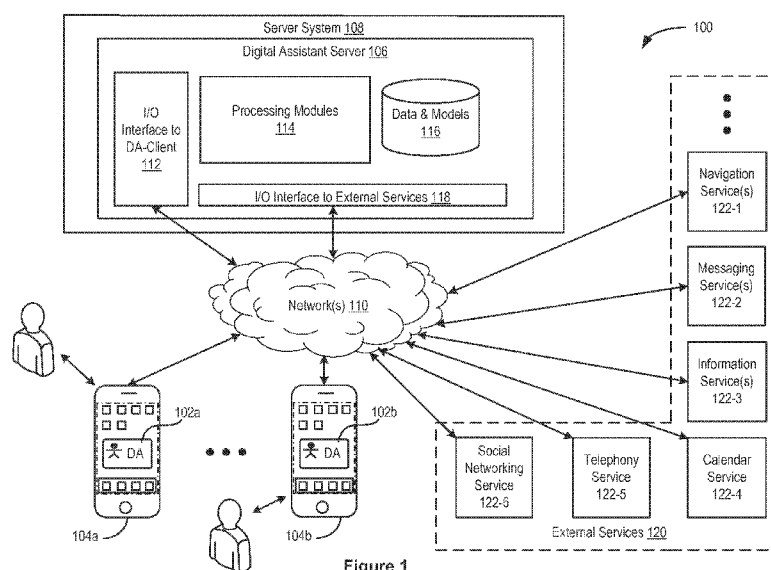


Figure 1

(57) **Abstract:** Methods and systems for recognizing textual identifiers within a plurality of words are described. A textual representation of a voice input is received from a user. The textual representation includes a plurality of words. A keyword is identified in the textual representation. It is determined whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers. Responsive to a determination that the one or more adjacent words correspond to a textual identifier, the keyword and the one or more adjacent words are replaced with the textual identifier.

SYSTEMS AND METHODS FOR RECOGNIZING TEXTUAL IDENTIFIERS WITHIN A PLURALITY OF WORDS

TECHNICAL FIELD

[0001] The disclosed embodiments relate generally to digital assistant systems, and more specifically, methods of recognizing textual identifiers in textual representations that are input by a user.

BACKGROUND

[0002] Just like human personal assistants, digital assistant systems can perform requested tasks and provide requested advice, information, or services. A digital assistant system's ability to fulfill a user's request is dependent on the digital assistant system's correct comprehension of the request or instructions. Recent advances in natural language processing have enabled users to interact with digital assistant systems using natural language, in spoken or textual forms. Such digital assistant systems can interpret the user's input to deduce the user's intent, translate the deduced intent into actionable tasks and parameters, execute operations or deploy services to perform the tasks, and produce output that is intelligible to the user. The ability of a digital assistant system to produce satisfactory responses to user requests depends on the natural language processing, knowledge base, and artificial intelligence available to the digital assistant system.

[0003] Also, users are increasingly using mobile devices to post status updates, messages, or blog posts to online services such as social networks, blogs, and the like. Traditionally, however, speech-to-text systems and/or digital assistants have been confined to information retrieval (e.g., web search), transcribing voice inputs for email or text messages, and the like, and have not been able to handle the special types of text that are sometimes used in postings to social networks, such as FACEBOOK or TWITTER. For example, a user may wish to input via a digital assistant special types of text, such as online handles or usernames (e.g., a TWITTER username), tags (e.g., a TWITTER hashtag), etc., that are difficult for traditional speech-to-text systems and/or digital assistants to identify as anything other than simple text. Accordingly, there is a need for digital assistants to be able to recognize when a user intends to input these special types of text via voice input, and process them appropriately.

SUMMARY

[0004] As described above, there is a need for digital assistant systems that are capable of recognizing special types of text, and processing them appropriately. This enables the digital assistant to provide a complete and comprehensive experience in contexts where special types of text are provided via voice input. For example, to enable digital assistants to provide a comprehensive user experience when posting to social networks, it is necessary to be able to recognize when a user is attempting to input a textual identifier (e.g., a TWITTER hashtag or username). This is especially helpful where it would otherwise be ambiguous whether a voice input should be transcribed directly, or converted or modified in some way to account for the user's intention. Specifically, a speech-to-text system will simply transcribe words, without recognizing the intent that they should have any special formatting: the spoken utterance "hashtag favorite band" will be transcribed just as it is spoken, rather than as the intended "#FavoriteBand." The disclosed systems and methods enable digital assistant systems to recognize when a user intends to input a textual identifier via a voice input, and replace it with the appropriate text, symbols, and/or formatting.

[0005] Some embodiments provide a method of recognizing textual identifiers within a plurality of words. The method is performed at an electronic device having one or more processors and memory storing one or more programs. The method includes receiving a textual representation of a user's voice input. In some embodiments, the voice input is received from the user. The textual representation including a plurality of words. The method further includes identifying a keyword in the textual representation. The method further includes determining whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers. The method further includes, responsive to a determination that the one or more adjacent words correspond to a textual identifier, replacing the keyword and the one or more adjacent words with the textual identifier. In some embodiments, the method further includes, responsive to a determination that the one or more adjacent words do not correspond to a textual identifier, not replacing the keyword and the one or more adjacent words.

[0006] In some embodiments, the adjacent words immediately follow the keyword. In some embodiments, at least one of the one or more adjacent words is composed only of a single letter. In some embodiments, the textual identifier is a concatenation of the one or

more adjacent words without interstitial spaces. In some embodiments, the textual identifier is a concatenation of a symbol and the one or more adjacent words without interstitial spaces.

[0007] In some embodiments, the textual identifier is a hashtag. In some embodiments, the hashtag is a TWITTER hashtag. In some embodiments, the hashtag includes the symbol “#” followed by one or more words.

[0008] In some embodiments, the collection of textual identifiers includes hashtags that were previously identified by a social network. In some embodiments, the collection includes hashtags that meet a predefined popularity metric. In some embodiments, the predefined popularity metric is a frequency of appearance in a social network within a predefined time period.

[0009] In some embodiments, the textual identifier is a username. In some embodiments, the username is a TWITTER username. In some embodiments, the username includes the symbol “@” followed by one or more words.

[0010] In some embodiments, the collection of textual identifiers includes usernames that are registered in a social network. In some embodiments, the social network is TWITTER.

[0011] In some embodiments, the collection of textual identifiers is a contact list of the user. In some embodiments, the collection is a set of usernames previously input by the user. In some embodiments, the textual identifier is an email address.

[0012] Some embodiments provide a method of recognizing textual identifiers within a plurality of words. The method is performed at an electronic device having one or more processors and memory storing one or more programs. The method includes receiving a textual representation of a user’s voice input, the textual representation including a plurality of words. In some embodiments, the method includes receiving the voice input from the user. The method further includes identifying a first keyword in the textual representation, the first keyword indicating the beginning of a textual tag, and identifying a second keyword in the textual representation, the second keyword indicating the end of a textual tag. The method further includes identifying one or more words between the first keyword and the second keyword. The method further includes replacing the first keyword, the second keyword, and the one or more words therebetween with a textual tag, wherein the textual tag comprises the one or more words.

[0013] In some embodiments, the textual tag further comprises a symbol preceding the one or more words. In some embodiments, the symbol is “#.” In some embodiments, the first keyword is “hashtag,” and the second keyword is “end hashtag.” In some embodiments, the textual identifier is a concatenation of the one or more adjacent words without interstitial spaces.

[0014] In accordance with some embodiments, a computer-readable storage medium (e.g., a non-transitory computer readable storage medium) is provided, the computer-readable storage medium storing one or more programs for execution by one or more processors of an electronic device, the one or more programs including instructions for performing any of the methods described above.

[0015] In accordance with some embodiments, an electronic device is provided that comprises means for performing any of the methods described above.

[0016] In accordance with some embodiments, an electronic device is provided that comprises a processing unit configured to perform any of the methods described above.

[0017] In accordance with some embodiments, an electronic device is provided that comprises one or more processors and memory storing one or more programs for execution by the one or more processors, the one or more programs including instructions for performing any of the methods described above.

[0018] In accordance with some embodiments, an information processing apparatus for use in an electronic device is provided, the information processing apparatus comprising means for performing any of the methods described above.

[0019] In accordance with some embodiments, a graphical user interface on a portable electronic device or a computer system with a display, a memory, and one or more processors to execute one or more programs stored in the memory is provided, the graphical user interface comprising user interfaces displayed in accordance with any of the methods described above.

[0020] Thus, digital assistant systems are provided with new and improved methods that recognize textual identifiers within a plurality of words. Such methods and systems may complement or replace existing methods and systems.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0021] Figure 1 is a block diagram illustrating an environment in which a digital assistant operates in accordance with some embodiments.
- [0022] Figure 2 is a block diagram illustrating a digital assistant client system in accordance with some embodiments.
- [0023] Figure 3A is a block diagram illustrating a standalone digital assistant system or a digital assistant server system in accordance with some embodiments.
- [0024] Figure 3B is a block diagram illustrating functions of the digital assistant shown in Figure 3A in accordance with some embodiments.
- [0025] Figure 3C is a network diagram illustrating a portion of an ontology in accordance with some embodiments.
- [0026] Figures 4A-5C are block diagrams illustrating example user interfaces depicting portions of dialogue between a digital assistant and a user in accordance with some embodiments.
- [0027] Figures 6-7 are flow charts illustrating methods of recognizing textual identifiers within a plurality of words in accordance with some embodiments.
- [0028] Figure 8 is a functional block diagram of an electronic device in accordance with some embodiments.
- [0029] Like reference numerals refer to corresponding parts throughout the drawings.

DESCRIPTION OF EMBODIMENTS

Digital Assistant

[0030] Figure 1 is a block diagram of an operating environment 100 of a digital assistant according to some embodiments. The terms “digital assistant,” “virtual assistant,” “intelligent automated assistant,” or “automatic digital assistant,” refer to any information processing system that interprets natural language input in spoken and/or textual form to deduce user intent (e.g., identify a task type that corresponds to the natural language input), and performs actions based on the deduced user intent (e.g., perform a task corresponding to the identified task type). For example, to act on a deduced user intent, the system can perform one or more of the following: identifying a task flow with steps and parameters designed to accomplish the deduced user intent (e.g., identifying a task type), inputting

specific requirements from the deduced user intent into the task flow, executing the task flow by invoking programs, methods, services, APIs, or the like (e.g., sending a request to a service provider); and generating output responses to the user in an audible (e.g., speech) and/or visual form.

[0031] Specifically, a digital assistant system is capable of accepting a user request at least partially in the form of a natural language command, request, statement, narrative, and/or inquiry. Typically, the user request seeks either an informational answer or performance of a task by the digital assistant system. A satisfactory response to the user request is generally either provision of the requested informational answer, performance of the requested task, or a combination of the two. For example, a user may ask the digital assistant system a question, such as “Where am I right now?” Based on the user’s current location, the digital assistant may answer, “You are in Central Park near the west gate.” The user may also request the performance of a task, for example, by stating “Please invite my friends to my girlfriend’s birthday party next week.” In response, the digital assistant may acknowledge the request by generating a voice output, “Yes, right away,” and then sending a suitable calendar invite from the user’s email address to each of the user’s friends listed in the user’s electronic address book or contact list. There are numerous other ways of interacting with a digital assistant to request information or performance of various tasks. In addition to providing verbal responses and taking programmed actions, the digital assistant can also provide responses in other visual or audio forms (e.g., as text, alerts, music, videos, animations, etc.).

[0032] As shown in Figure 1, in some embodiments, a digital assistant system is implemented according to a client-server model. The digital assistant system includes a client-side portion (e.g., 102a and 102b) (hereafter “digital assistant (DA) client 102”) executed on a user device (e.g., 104a and 104b), and a server-side portion 106 (hereafter “digital assistant (DA) server 106”) executed on a server system 108. The DA client 102 communicates with the DA server 106 through one or more networks 110. The DA client 102 provides client-side functionalities such as user-facing input and output processing and communications with the DA server 106. The DA server 106 provides server-side functionalities for any number of DA clients 102 each residing on a respective user device 104 (also called a client device).

[0033] In some embodiments, the DA server 106 includes a client-facing I/O interface 112, one or more processing modules 114, data and models 116, and an I/O interface to external services 118. The client-facing I/O interface facilitates the client-facing input and output processing for the digital assistant server 106. The one or more processing modules 114 utilize the data and models 116 to determine the user's intent based on natural language input and perform task execution based on the deduced user intent.

[0034] In some embodiments, the DA server 106 communicates with external services 120 (e.g., navigation service(s) 122-1, messaging service(s) 122-2, information service(s) 122-3, calendar service 122-4, telephony service 122-5, social networking service 122-6, etc.) through the network(s) 110 for task completion or information acquisition. In some embodiments, the I/O interface to the external services 118 facilitates such communications. In some embodiments, the DA client 102 communicates with external services 120 through the network(s) 110 for task completion or information acquisition.

[0035] Examples of the user device 104 include, but are not limited to, a handheld computer, a personal digital assistant (PDA), a tablet computer, a laptop computer, a desktop computer, a cellular telephone, a smartphone, an enhanced general packet radio service (EGPRS) mobile phone, a media player, a navigation device, a game console, a television, a remote control, or a combination of any two or more of these data processing devices or any other suitable data processing devices. More details on the user device 104 are provided in reference to an exemplary user device 104 shown in Figure 2.

[0036] Examples of the communication network(s) 110 include local area networks ("LAN") and wide area networks ("WAN"), e.g., the Internet. The communication network(s) 110 may be implemented using any known network protocol, including various wired or wireless protocols, such as Ethernet, Universal Serial Bus (USB), FIREWIRE, Global System for Mobile Communications (GSM), Enhanced Data GSM Environment (EDGE), code division multiple access (CDMA), time division multiple access (TDMA), Bluetooth, Wi-Fi, voice over Internet Protocol (VoIP), Wi-MAX, or any other suitable communication protocol.

[0037] The server system 108 can be implemented on at least one data processing apparatus and/or a distributed network of computers. In some embodiments, the server system 108 also employs various virtual devices and/or services of third party service

providers (e.g., third-party cloud service providers) to provide the underlying computing resources and/or infrastructure resources of the server system 108.

[0038] Although the digital assistant system shown in Figure 1 includes both a client-side portion (e.g., the DA client 102) and a server-side portion (e.g., the DA server 106), in some embodiments, a digital assistant system refers only to the server-side portion (e.g., the DA server 106). In some embodiments, the functions of a digital assistant can be implemented as a standalone application installed on a user device. In addition, the divisions of functionalities between the client and server portions of the digital assistant can vary in different embodiments. For example, in some embodiments, the DA client 102 is a thin-client that provides only user-facing input and output processing functions, and delegates all other functionalities of the digital assistant to the DA server 106. In some other embodiments, the DA client 102 is configured to perform or assist one or more functions of the DA server 106.

[0039] Figure 2 is a block diagram of a user device 104 in accordance with some embodiments. The user device 104 includes a memory interface 202, one or more processors 204, and a peripherals interface 206. The various components in the user device 104 are coupled by one or more communication buses or signal lines. The user device 104 includes various sensors, subsystems, and peripheral devices that are coupled to the peripherals interface 206. The sensors, subsystems, and peripheral devices gather information and/or facilitate various functionalities of the user device 104.

[0040] For example, in some embodiments, a motion sensor 210, a light sensor 212, and a proximity sensor 214 are coupled to the peripherals interface 206 to facilitate orientation, light, and proximity sensing functions. In some embodiments, other sensors 216, such as a positioning system (e.g., GPS receiver), a temperature sensor, a biometric sensor, and the like, are connected to the peripherals interface 206, to facilitate related functionalities.

[0041] In some embodiments, the user device 104 includes a camera subsystem 220 coupled to the peripherals interface 206. In some embodiments, an optical sensor 222 of the camera subsystem 220 facilitates camera functions, such as taking photographs and recording video clips. In some embodiments, the user device 104 includes one or more wired and/or wireless communication subsystems 224 to provide communication functions. The communication subsystems 224 typically include various communication ports, radio frequency receivers and transmitters, and/or optical (e.g., infrared) receivers and transmitters.

In some embodiments, the user device 104 includes an audio subsystem 226 coupled to one or more speakers 228 and one or more microphones 230 to facilitate voice-enabled functions, such as voice recognition, voice transcription, voice replication, digital recording, and telephony functions.

[0042] In some embodiments, an I/O subsystem 240 is also coupled to the peripherals interface 206. In some embodiments, the user device 104 includes a touch screen 246, and the I/O subsystem 240 includes a touch screen controller 242 coupled to the touch screen 246. When the user device 104 includes the touch screen 246 and the touch screen controller 242, the touch screen 246 and the touch screen controller 242 are typically configured to, for example, detect contact and movement or break thereof using any of a plurality of touch sensitivity technologies, such as capacitive, resistive, infrared, surface acoustic wave technologies, proximity sensor arrays, and the like. In some embodiments, the user device 104 includes a display that does not include a touch-sensitive surface. In some embodiments, the user device 104 includes a separate touch-sensitive surface. In some embodiments, the user device 104 includes other input controller(s) 244. When the user device 104 includes the other input controller(s) 244, the other input controller(s) 244 are typically coupled to other input/control devices 248, such as one or more buttons, rocker switches, thumb-wheels, infrared ports, USB ports, and/or pointer devices (such as a stylus).

[0043] The memory interface 202 is coupled to memory 250. In some embodiments, the memory 250 includes a non-transitory computer readable medium, such as high-speed random access memory and/or non-volatile memory (e.g., one or more magnetic disk storage devices, one or more flash memory devices, one or more optical storage devices, and/or other non-volatile solid-state memory devices).

[0044] In some embodiments, the memory 250 stores an operating system 252, a communications module 254, a graphical user interface (“GUI”) module 256, a sensor processing module 258, a phone module 260, and applications 262, and a subset or superset thereof. The operating system 252 includes instructions for handling basic system services and for performing hardware dependent tasks. The communications module 254 facilitates communicating with one or more additional devices, one or more computers and/or one or more servers. The graphical user interface module 256 facilitates graphic user interface processing. The sensor processing module 258 facilitates sensor-related processing and functions (e.g., processing voice input received with the one or more microphones 228). The

phone module 260 facilitates phone-related processes and functions. The application module 262 facilitates various functionalities of user applications, such as electronic-messaging, web browsing, media processing, navigation, imaging and/or other processes and functions. In some embodiments, the user device 104 stores in the memory 250 one or more software applications 270-1 and 270-2 each associated with at least one of the external service providers.

[0045] As described above, in some embodiments, the memory 250 also stores client-side digital assistant instructions (e.g., in a digital assistant client module 264) and various user data 266 (e.g., user-specific vocabulary data, preference data, and/or other data such as the user's electronic address book or contact list, to-do lists, shopping lists, etc.) to provide the client-side functionalities of the digital assistant.

[0046] In various embodiments, the digital assistant client module 264 is capable of accepting voice input, text input, touch input, and/or gestural input through various user interfaces (e.g., the I/O subsystem 240) of the user device 104. The digital assistant client module 264 is also capable of providing output in audio, visual, and/or tactile forms. For example, output can be provided as voice, sound, alerts, text messages, menus, graphics, videos, animations, vibrations, and/or combinations of two or more of the above. During operation, the digital assistant client module 264 communicates with the digital assistant server (e.g., the digital assistant server 106, Figure 1) using the communication subsystems 224.

[0047] In some embodiments, the digital assistant client module 264 utilizes various sensors, subsystems and peripheral devices to gather additional information from the surrounding environment of the user device 104 to establish a context associated with a user input. In some embodiments, the digital assistant client module 264 provides the context information or a subset thereof with the user input to the digital assistant server (e.g., the digital assistant server 106, Figure 1) to help deduce the user's intent.

[0048] In some embodiments, the context information that can accompany the user input includes sensor information, *e.g.*, lighting, ambient noise, ambient temperature, images or videos of the surrounding environment, etc. In some embodiments, the context information also includes the physical state of the device, *e.g.*, device orientation, device location, device temperature, power level, speed, acceleration, motion patterns, cellular signal strength, etc. In some embodiments, information related to the software state of the user

device 104, *e.g.*, running processes, installed programs, past and present network activities, background services, error logs, resource usage, etc., of the user device 104 is also provided to the digital assistant server (*e.g.*, the digital assistant server 106, Figure 1) as context information associated with a user input.

[0049] In some embodiments, the DA client module 264 selectively provides information (*e.g.*, at least a portion of the user data 266) stored on the user device 104 in response to requests from the digital assistant server. In some embodiments, the digital assistant client module 264 also elicits additional input from the user via a natural language dialogue or other user interfaces upon request by the digital assistant server 106 (Figure 1). The digital assistant client module 264 passes the additional input to the digital assistant server 106 to help the digital assistant server 106 in intent deduction and/or fulfillment of the user's intent expressed in the user request.

[0050] In some embodiments, the memory 250 may include additional instructions or fewer instructions. Furthermore, various functions of the user device 104 may be implemented in hardware and/or in firmware, including in one or more signal processing and/or application specific integrated circuits, and the user device 104, thus, need not include all modules and applications illustrated in Figure 2.

[0051] Figure 3A is a block diagram of an exemplary digital assistant system 300 in accordance with some embodiments. In some embodiments, the digital assistant system 300 is implemented on a standalone computer system. In some embodiments, the digital assistant system 300 is distributed across multiple computers. In some embodiments, some of the modules and functions of the digital assistant are divided into a server portion and a client portion, where the client portion resides on a user device (*e.g.*, the user device 104) and communicates with the server portion (*e.g.*, the server system 108) through one or more networks, *e.g.*, as shown in Figure 1. In some embodiments, the digital assistant system 300 is an embodiment of the server system 108 (and/or the digital assistant server 106) shown in Figure 1. In some embodiments, the digital assistant system 300 is implemented in a user device (*e.g.*, the user device 104, Figure 1), thereby eliminating the need for a client-server system. It should be noted that the digital assistant system 300 is only one example of a digital assistant system, and that the digital assistant system 300 may have more or fewer components than shown, may combine two or more components, or may have a different configuration or arrangement of the components. The various components shown in Figure

3A may be implemented in hardware, software, firmware, including one or more signal processing and/or application specific integrated circuits, or a combination of thereof.

[0052] The digital assistant system 300 includes memory 302, one or more processors 304, an input/output (I/O) interface 306, and a network communications interface 308. These components communicate with one another over one or more communication buses or signal lines 310.

[0053] In some embodiments, the memory 302 includes a non-transitory computer readable medium, such as high-speed random access memory and/or a non-volatile computer readable storage medium (e.g., one or more magnetic disk storage devices, one or more flash memory devices, one or more optical storage devices, and/or other non-volatile solid-state memory devices).

[0054] The I/O interface 306 couples input/output devices 316 of the digital assistant system 300, such as displays, a keyboards, touch screens, and microphones, to the user interface module 322. The I/O interface 306, in conjunction with the user interface module 322, receives user inputs (e.g., voice input, keyboard inputs, touch inputs, etc.) and process them accordingly. In some embodiments, when the digital assistant is implemented on a standalone user device, the digital assistant system 300 includes any of the components and I/O and communication interfaces described with respect to the user device 104 in Figure 2 (e.g., one or more microphones 230). In some embodiments, the digital assistant system 300 represents the server portion of a digital assistant implementation, and interacts with the user through a client-side portion residing on a user device (e.g., the user device 104 shown in Figure 2).

[0055] In some embodiments, the network communications interface 308 includes wired communication port(s) 312 and/or wireless transmission and reception circuitry 314. The wired communication port(s) receive and send communication signals via one or more wired interfaces, e.g., Ethernet, Universal Serial Bus (USB), FIREWIRE, etc. The wireless circuitry 314 typically receives and sends RF signals and/or optical signals from/to communications networks and other communications devices. The wireless communications may use any of a plurality of communications standards, protocols and technologies, such as GSM, EDGE, CDMA, TDMA, Bluetooth, Wi-Fi, VoIP, Wi-MAX, or any other suitable communication protocol. The network communications interface 308 enables communication between the digital assistant system 300 with networks, such as the Internet,

an intranet and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN) and/or a metropolitan area network (MAN), and other devices.

[0056] In some embodiments, the non-transitory computer readable storage medium of memory 302 stores programs, modules, instructions, and data structures including all or a subset of: an operating system 318, a communications module 320, a user interface module 322, one or more applications 324, and a digital assistant module 326. The one or more processors 304 execute these programs, modules, and instructions, and reads/writes from/to the data structures.

[0057] The operating system 318 (e.g., Darwin, RTXC, LINUX, UNIX, OS X, iOS, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communications between various hardware, firmware, and software components.

[0058] The communications module 320 facilitates communications between the digital assistant system 300 with other devices over the network communications interface 308. For example, the communication module 320 may communicate with the communications module 254 of the device 104 shown in Figure 2. The communications module 320 also includes various software components for handling data received by the wireless circuitry 314 and/or wired communications port 312.

[0059] In some embodiments, the user interface module 322 receives commands and/or inputs from a user via the I/O interface 306 (e.g., from a keyboard, touch screen, and/or microphone), and provides user interface objects on a display.

[0060] The applications 324 include programs and/or modules that are configured to be executed by the one or more processors 304. For example, if the digital assistant system is implemented on a standalone user device, the applications 324 may include user applications, such as games, a calendar application, a navigation application, or an email application. If the digital assistant system 300 is implemented on a server farm, the applications 324 may include resource management applications, diagnostic applications, or scheduling applications, for example.

[0061] The memory 302 also stores the digital assistant module (or the server portion of a digital assistant) 326. In some embodiments, the digital assistant module 326 includes the following sub-modules, or a subset or superset thereof: an input/output processing

module 328, a speech-to-text (STT) processing module 330, a natural language processing module 332, a dialogue flow processing module 334, a task flow processing module 336, and a service processing module 338. Each of these processing modules has access to one or more of the following data and models of the digital assistant 326, or a subset or superset thereof: ontology 360, vocabulary index 344, user data 348, task flow models 354, and service models 356.

[0062] In some embodiments, using the processing modules (e.g., the input/output processing module 328, the STT processing module 330, the natural language processing module 332, the dialogue flow processing module 334, the task flow processing module 336, and/or the service processing module 338), data, and models implemented in the digital assistant module 326, the digital assistant system 300 performs at least some of the following: identifying a user's intent expressed in a natural language input received from the user; actively eliciting and obtaining information needed to fully deduce the user's intent (e.g., by disambiguating words, names, intentions, etc.); determining the task flow for fulfilling the deduced intent; and executing the task flow to fulfill the deduced intent. In some embodiments, the digital assistant also takes appropriate actions when a satisfactory response was not or could not be provided to the user for various reasons.

[0063] As shown in Figure 3B, in some embodiments, the I/O processing module 328 interacts with the user through the I/O devices 316 in Figure 3A or with a user device (e.g., a user device 104 in Figure 1) through the network communications interface 308 in Figure 3A to obtain user input (e.g., a speech input) and to provide responses to the user input. The I/O processing module 328 optionally obtains context information associated with the user input from the user device, along with or shortly after the receipt of the user input. The context information includes user-specific data, vocabulary, and/or preferences relevant to the user input. In some embodiments, the context information also includes software and hardware states of the device (e.g., the user device 104 in Figure 1) at the time the user request is received, and/or information related to the surrounding environment of the user at the time that the user request was received. In some embodiments, the I/O processing module 328 also sends follow-up questions to, and receives answers from, the user regarding the user request. In some embodiments, when a user request is received by the I/O processing module 328 and the user request contains a speech input, the I/O processing module 328 forwards the speech input to the speech-to-text (STT) processing module 330 for speech-to-text conversions.

[0064] In some embodiments, the speech-to-text processing module 330 receives speech input (e.g., a user utterance captured in a voice recording) through the I/O processing module 328. In some embodiments, the speech-to-text processing module 330 uses various acoustic and language models to recognize the speech input as a sequence of phonemes, and ultimately, a sequence of words or tokens written in one or more languages. The speech-to-text processing module 330 is implemented using any suitable speech recognition techniques, acoustic models, and language models, such as Hidden Markov Models, Dynamic Time Warping (DTW)-based speech recognition, and other statistical and/or analytical techniques. In some embodiments, the speech-to-text processing can be performed at least partially by a third party service or on the user's device. Once the speech-to-text processing module 330 obtains the result of the speech-to-text processing (e.g., a sequence of words or tokens), it passes the result to the natural language processing module 332 for intent deduction.

[0065] The natural language processing module 332 ("natural language processor") of the digital assistant 326 takes the sequence of words or tokens ("token sequence") generated by the speech-to-text processing module 330, and attempts to associate the token sequence with one or more "actionable intents" recognized by the digital assistant. As used herein, an "actionable intent" represents a task that can be performed by the digital assistant 326 and/or the digital assistant system 300 (Figure 3A), and has an associated task flow implemented in the task flow models 354. The associated task flow is a series of programmed actions and steps that the digital assistant system 300 takes in order to perform the task. The scope of a digital assistant system's capabilities is dependent on the number and variety of task flows that have been implemented and stored in the task flow models 354, or in other words, on the number and variety of "actionable intents" that the digital assistant system 300 recognizes. The effectiveness of the digital assistant system 300, however, is also dependent on the digital assistant system's ability to deduce the correct "actionable intent(s)" from the user request expressed in natural language.

[0066] In some embodiments, in addition to the sequence of words or tokens obtained from the speech-to-text processing module 330, the natural language processor 332 also receives context information associated with the user request (e.g., from the I/O processing module 328). The natural language processor 332 optionally uses the context information to clarify, supplement, and/or further define the information contained in the token sequence received from the speech-to-text processing module 330. The context information includes, for example, user preferences, hardware and/or software states of the user device, sensor

information collected before, during, or shortly after the user request, prior interactions (*e.g.*, dialogue) between the digital assistant and the user, and the like.

[0067] In some embodiments, the natural language processing is based on an ontology 360. The ontology 360 is a hierarchical structure containing a plurality of nodes, each node representing either an “actionable intent” or a “property” relevant to one or more of the “actionable intents” or other “properties.” As noted above, an “actionable intent” represents a task that the digital assistant system 300 is capable of performing (*e.g.*, a task that is “actionable” or can be acted on). A “property” represents a parameter associated with an actionable intent or a sub-aspect of another property. A linkage between an actionable intent node and a property node in the ontology 360 defines how a parameter represented by the property node pertains to the task represented by the actionable intent node.

[0068] In some embodiments, the ontology 360 is made up of actionable intent nodes and property nodes. Within the ontology 360, each actionable intent node is linked to one or more property nodes either directly or through one or more intermediate property nodes. Similarly, each property node is linked to one or more actionable intent nodes either directly or through one or more intermediate property nodes. For example, the ontology 360 shown in Figure 3C includes a “restaurant reservation” node, which is an actionable intent node. Property nodes “restaurant,” “date/time” (for the reservation), and “party size” are each directly linked to the “restaurant reservation” node (*i.e.*, the actionable intent node). In addition, property nodes “cuisine,” “price range,” “phone number,” and “location” are sub-nodes of the property node “restaurant,” and are each linked to the “restaurant reservation” node (*i.e.*, the actionable intent node) through the intermediate property node “restaurant.” For another example, the ontology 360 shown in Figure 3C also includes a “set reminder” node, which is another actionable intent node. Property nodes “date/time” (for the setting the reminder) and “subject” (for the reminder) are each linked to the “set reminder” node. Since the property “date/time” is relevant to both the task of making a restaurant reservation and the task of setting a reminder, the property node “date/time” is linked to both the “restaurant reservation” node and the “set reminder” node in the ontology 360.

[0069] An actionable intent node, along with its linked concept nodes, may be described as a “domain.” In the present discussion, each domain is associated with a respective actionable intent, and refers to the group of nodes (and the relationships therebetween) associated with the particular actionable intent. For example, the ontology 360

shown in Figure 3C includes an example of a restaurant reservation domain 362, a reminder domain 364, and a social network domain 366 within the ontology 360. The restaurant reservation domain includes the actionable intent node “restaurant reservation,” property nodes “restaurant,” “date/time,” and “party size,” and sub-property nodes “cuisine,” “price range,” “phone number,” and “location.” The reminder domain 364 includes the actionable intent node “set reminder,” and property nodes “subject” and “date/time.” The social network domain 366 includes the actionable intent node “post to a social network,” and property nodes “social network ID” and “message body.” In some embodiments, the ontology 360 is made up of many domains. Each domain may share one or more property nodes with one or more other domains. For example, the “date/time” property node may be associated with many other domains (e.g., a scheduling domain, a travel reservation domain, a movie ticket domain, etc.), in addition to the restaurant reservation domain 362 and the reminder domain 364.

[0070] While Figure 3C illustrates three exemplary domains within the ontology 360, the ontology 360 may include other domains (or actionable intents), such as “initiate a phone call,” “find directions,” “schedule a meeting,” “send a message,” “provide an answer to a question,” and so on. For example, a “send a message” domain is associated with a “send a message” actionable intent node, and may further include property nodes such as “recipient(s),” “message type,” and “message body.” The property node “recipient” may be further defined, for example, by the sub-property nodes such as “recipient name” and “message address.”

[0071] In some embodiments, the ontology 360 includes all of the domains (and hence actionable intents) that the digital assistant is capable of understanding and acting upon. In some embodiments, the ontology 360 may be modified, such as by adding or removing domains or nodes, or by modifying relationships between the nodes within the ontology 360.

[0072] In some embodiments, nodes associated with multiple related actionable intents may be clustered under a “super domain” in the ontology 360. For example, a “travel” super-domain may include a cluster of property nodes and actionable intent nodes related to travel. The actionable intent nodes related to travel may include “airline reservation,” “hotel reservation,” “car rental,” “get directions,” “find points of interest,” and so on. The actionable intent nodes under the same super domain (e.g., the “travel” super

domain) may have many property nodes in common. For example, the actionable intent nodes for “airline reservation,” “hotel reservation,” “car rental,” “get directions,” “find points of interest” may share one or more of the property nodes “start location,” “destination,” “departure date/time,” “arrival date/time,” and “party size.”

[0073] In some embodiments, each node in the ontology 360 is associated with a set of words and/or phrases that are relevant to the property or actionable intent represented by the node. The respective set of words and/or phrases associated with each node is the so-called “vocabulary” associated with the node. The respective set of words and/or phrases associated with each node can be stored in the vocabulary index 344 (Figure 3B) in association with the property or actionable intent represented by the node. For example, returning to Figure 3B, the vocabulary associated with the node for the property of “restaurant” may include words such as “food,” “drinks,” “cuisine,” “hungry,” “eat,” “pizza,” “fast food,” “meal,” and so on. For another example, the vocabulary associated with the node for the actionable intent of “initiate a phone call” may include words and phrases such as “call,” “phone,” “dial,” “ring,” “call this number,” “make a call to,” and so on. The vocabulary index 344 optionally includes words and phrases in different languages.

[0074] In some embodiments, the natural language processor 332 shown in Figure 3B receives the token sequence (e.g., a text string) from the speech-to-text processing module 330, and determines what nodes are implicated by the words in the token sequence. In some embodiments, if a word or phrase in the token sequence is found to be associated with one or more nodes in the ontology 360 (via the vocabulary index 344), the word or phrase will “trigger” or “activate” those nodes. When multiple nodes are “triggered,” based on the quantity and/or relative importance of the activated nodes, the natural language processor 332 will select one of the actionable intents as the task (or task type) that the user intended the digital assistant to perform. In some embodiments, the domain that has the most “triggered” nodes is selected. In some embodiments, the domain having the highest confidence value (e.g., based on the relative importance of its various triggered nodes) is selected. In some embodiments, the domain is selected based on a combination of the number and the importance of the triggered nodes. In some embodiments, additional factors are considered in selecting the node as well, such as whether the digital assistant system 300 has previously correctly interpreted a similar request from a user.

[0075] In some embodiments, the digital assistant system 300 also stores names of specific entities in the vocabulary index 344, so that when one of these names is detected in the user request, the natural language processor 332 will be able to recognize that the name refers to a specific instance of a property or sub-property in the ontology. In some embodiments, the names of specific entities are names of businesses, restaurants, people, movies, and the like. In some embodiments, the digital assistant system 300 can search and identify specific entity names from other data sources, such as the user's address book or contact list, a movies database, a musicians database, and/or a restaurant database. In some embodiments, when the natural language processor 332 identifies that a word in the token sequence is a name of a specific entity (such as a name in the user's address book or contact list), that word is given additional significance in selecting the actionable intent within the ontology for the user request.

[0076] For example, when the words "Mr. Santo" are recognized from the user request, and the last name "Santo" is found in the vocabulary index 344 as one of the contacts in the user's contact list, then it is likely that the user request corresponds to a "send a message" or "initiate a phone call" domain. For another example, when the words "ABC Café" are found in the user request, and the term "ABC Café" is found in the vocabulary index 344 as the name of a particular restaurant in the user's city, then it is likely that the user request corresponds to a "restaurant reservation" domain.

[0077] User data 348 includes user-specific information, such as user-specific vocabulary, user preferences, user address, user's default and secondary languages, user's contact list, and other short-term or long-term information for each user. The natural language processor 332 can use the user-specific information to supplement the information contained in the user input to further define the user intent. For example, for a user request "invite my friends to my birthday party," the natural language processor 332 is able to access user data 348 to determine who the "friends" are and when and where the "birthday party" would be held, rather than requiring the user to provide such information explicitly in his/her request.

[0078] Once the natural language processor 332 identifies an actionable intent (or domain) based on the user request, the natural language processor 332 generates a structured query to represent the identified actionable intent. In some embodiments, the structured query includes parameters for one or more nodes within the domain for the actionable intent,

and at least some of the parameters are populated with the specific information and requirements specified in the user request. For example, the user may say “Make me a dinner reservation at a sushi place at 7.” In this case, the natural language processor 332 may be able to correctly identify the actionable intent to be “restaurant reservation” based on the user input. According to the ontology, a structured query for a “restaurant reservation” domain may include parameters such as {Cuisine}, {Time}, {Date}, {Party Size}, and the like. Based on the information contained in the user’s utterance, the natural language processor 332 may generate a partial structured query for the restaurant reservation domain, where the partial structured query includes the parameters {Cuisine= “Sushi”} and {Time = “7pm”}. However, in this example, the user’s utterance contains insufficient information to complete the structured query associated with the domain. Therefore, other necessary parameters such as {Party Size} and {Date} are not specified in the structured query based on the information currently available. In some embodiments, the natural language processor 332 populates some parameters of the structured query with received context information. For example, if the user requested a sushi restaurant “near me,” the natural language processor 332 may populate a {location} parameter in the structured query with GPS coordinates from the user device 104.

[0079] In some embodiments, the natural language processor 332 passes the structured query (including any completed parameters) to the task flow processing module 336 (“task flow processor”). The task flow processor 336 is configured to perform one or more of: receiving the structured query from the natural language processor 332, completing the structured query, and performing the actions required to “complete” the user’s ultimate request. In some embodiments, the various procedures necessary to complete these tasks are provided in task flow models 354. In some embodiments, the task flow models 354 include procedures for obtaining additional information from the user, and task flows for performing actions associated with the actionable intent.

[0080] As described above, in order to complete a structured query, the task flow processor 336 may need to initiate additional dialogue with the user in order to obtain additional information, and/or disambiguate potentially ambiguous utterances. When such interactions are necessary, the task flow processor 336 invokes the dialogue processing module 334 (“dialogue processor”) to engage in a dialogue with the user. In some embodiments, the dialogue processing module 334 determines how (and/or when) to ask the user for the additional information, and receives and processes the user responses. In some

embodiments, the questions are provided to and answers are received from the users through the I/O processing module 328. For example, the dialogue processing module 334 presents dialogue output to the user via audio and/or visual output, and receives input from the user via spoken or physical (e.g., touch gesture) responses. Continuing with the example above, when the task flow processor 336 invokes the dialogue processor 334 to determine the “party size” and “date” information for the structured query associated with the domain “restaurant reservation,” the dialogue processor 334 generates questions such as “For how many people?” and “On which day?” to pass to the user. Once answers are received from the user, the dialogue processing module 334 populates the structured query with the missing information, or passes the information to the task flow processor 336 to complete the missing information from the structured query.

[0081] In some cases, the task flow processor 336 may receive a structured query that has one or more ambiguous properties. For example, a structured query for the “send a message” domain may indicate that the intended recipient is “Bob,” and the user may have multiple contacts named “Bob.” The task flow processor 336 will request that the dialogue processor 334 disambiguate this property of the structured query. In turn, the dialogue processor 334 may ask the user “Which Bob?”, and display (or read) a list of contacts named “Bob” from which the user may choose.

[0082] Once the task flow processor 336 has completed the structured query for an actionable intent, the task flow processor 336 proceeds to perform the ultimate task associated with the actionable intent. Accordingly, the task flow processor 336 executes the steps and instructions in the task flow model according to the specific parameters contained in the structured query. For example, the task flow model for the actionable intent of “restaurant reservation” may include steps and instructions for contacting a restaurant and actually requesting a reservation for a particular party size at a particular time. For example, using a structured query such as: {restaurant reservation, restaurant = ABC Café, date = 3/12/2012, time = 7pm, party size = 5}, the task flow processor 336 may perform the steps of: (1) logging onto a server of the ABC Café or a restaurant reservation system that is configured to accept reservations for multiple restaurants, such as the ABC Café, (2) entering the date, time, and party size information in a form on the website, (3) submitting the form, and (4) making a calendar entry for the reservation in the user’s calendar.

[0083] In some embodiments, the task flow processor 336 employs the assistance of a service processing module 338 (“service processor”) to complete a task requested in the user input or to provide an informational answer requested in the user input. For example, the service processor 338 can act on behalf of the task flow processor 336 to make a phone call, set a calendar entry, invoke a map search, invoke or interact with other user applications installed on the user device, and invoke or interact with third party services (e.g. a restaurant reservation portal, a social networking website or service, a banking portal, etc.). In some embodiments, the protocols and application programming interfaces (API) required by each service can be specified by a respective service model among the service models 356. The service processor 338 accesses the appropriate service model for a service and generates requests for the service in accordance with the protocols and APIs required by the service according to the service model.

[0084] For example, if a restaurant has enabled an online reservation service, the restaurant can submit a service model specifying the necessary parameters for making a reservation and the APIs for communicating the values of the necessary parameters to the online reservation service. When requested by the task flow processor 336, the service processor 338 can establish a network connection with the online reservation service using the web address stored in the service models 356, and send the necessary parameters of the reservation (e.g., time, date, party size) to the online reservation interface in a format according to the API of the online reservation service.

[0085] For another example, a social networking service can submit a service model specifying the necessary parameters for posting a message to the service and the APIs for communicating the values of the necessary parameters to the social networking service. When requested by the task flow processor 336, the service processor 338 can establish a network connection with the social networking service using a web address stored in the service models 356, and send the necessary parameters of the message (e.g., account username, credentials, message content) to the social networking service in a format according to the established API.

[0086] In some embodiments, the natural language processor 332, dialogue processor 334, and task flow processor 336 are used collectively and iteratively to deduce and define the user’s intent, obtain information to further clarify and refine the user intent, and finally

generate a response (e.g., provide an output to the user, or complete a task) to fulfill the user's intent.

[0087] In some embodiments, after all of the tasks needed to fulfill the user's request have been performed, the digital assistant 326 formulates a confirmation response, and sends the response back to the user through the I/O processing module 328. If the user request seeks an informational answer, the confirmation response presents the requested information to the user. In some embodiments, the digital assistant also requests the user to indicate whether the user is satisfied with the response produced by the digital assistant 326.

Automatic Identification of Textual Identifiers

[0088] As noted above, digital assistants are sometimes used to quickly and easily create textual messages to be sent to a recipient (e.g., via text message, email, etc.), or posted to an information service (e.g., a social network, a blog, a website, etc.). For example, a user may tell a digital assistant to "Send Bob a text message saying I'm on my way," causing the digital assistant to send a text message (e.g., via short message service, or "SMS") with the content "I'm on my way." The user's utterance is, for example, transcribed by the speech-to-text processing module 330 of a digital assistant system 300. Once transcribed, the digital assistant uses the converted text to determine the user's intent (to send a text message) and the intended content of the message ("I'm on my way").

[0089] Increasingly, users are also using mobile devices to post status updates, messages, or blog posts to external services such as social networks, blogs, and the like. Many social networks provide mobile-device applications (e.g., included in applications 262) from which a user can input a message to be posted, broadcast, or otherwise made available to other users. For example, a FACEBOOK application may allow a user to post status updates to his FACEBOOK account. Likewise, a TWITTER application may allow a user to compose and post short text messages, or "tweets," which are then posted to or made available to other TWITTER users.

[0090] However, text input mechanisms for portable electronic devices, such as smart phones, tend to be very small, and inputting text can be cumbersome and time consuming. Accordingly, it would be advantageous to leverage the capabilities of a digital assistant, including its speech-to-text capabilities, to help simplify text input for messages and social network postings. Moreover, because the digital assistant can interact with external services, such as a social networking service 122-6, the digital assistant can post a user's text directly

to that external service without requiring the user to open a separate application associated with that service (e.g., a dedicated FACEBOOK or TWITTER application). In some embodiments, a digital assistant is permitted to post to a particular service if there is an application associated with that service installed on the same device as the digital assistant, or otherwise associated with a user of the digital assistant.

[0091] While a typical speech-to-text system may simplify text input on mobile devices, it will not allow a user to take advantage of all of the functionality provided by certain services or social networks. Specifically, some social networks allow special textual identifiers to differentiate and/or signify certain types of information within their postings. For example, a social networking service may use special characters to signify portions of the text that correspond to usernames or tags within a posting. The social network may use these characters to facilitate indexing, searching, and/or analyzing postings, and/or to differentiate (graphically or otherwise) the usernames or tags from the surrounding text.

[0092] One social networking service that uses special textual identifiers to signify certain types of information is TWITTER, where users compose short messages (e.g., containing 144 characters or less), and post those messages to their accounts. These messages, or “tweets,” are then available to other users within the social network. In many cases, a user’s tweets are sent to the accounts of anyone who has chosen to “follow” the user, rather than to a specified addressee (as in the case of email or text messages).

[0093] Two examples of special textual identifiers recognized in TWITTER messages are the username and the hashtag. TWITTER usernames uniquely identify members of the social network, and are preceded by the “@” symbol. For example, a user may be associated with a username “@user1234.” (The “@” symbol is typically verbalized as the word “at,” so this username may be spoken as “at user 1234.”) Usernames are often included in the text of a tweet, for example, when a user is responding to or referring to another user. Usernames can then be used to organize messages for searching, categorization, and the like, such as when someone wishes to search for tweets posted by a particular user, or that mention a particular user.

[0094] Hashtags are any combination of words, letters, numbers, and/or symbols that follow a “#” symbol (without interstitial spaces). As discussed below, hashtags can be used to organize messages for searching, categorization, and the like. In many cases, hashtags are contextually relevant to the tweet in which they are included. For example, a user may

compose a tweet saying “I like Mint Chocolate Chip! #FavoriteIceCream.” In this case, the hashtag “#FavoriteIceCream” is relevant to the content of the tweet. Often, hashtags become social trends, and numerous users will post tweets with the same hashtag in order to participate in the trend. For example, another user may read the above tweet regarding his friend’s preference for mint chocolate chip ice cream, and post a message with his preference: “Call me boring, but I love vanilla! #FavoriteIceCream.”

[0095] As noted above, usernames and hashtags can be used to facilitate indexing, searching, and analyzing messages posted to a social network, such as TWITTER. For example, a TWITTER user can easily search for tweets that mention a particular person by entering that person’s TWITTER username into a search field. Because usernames are preceded by the “@” symbol, searching for tweets based on usernames can be performed faster and more efficiently than by full text searching. Similarly, a user can search for tweets that refer to a particular subject matter by searching for an associated hashtag. Continuing the example from above, a user may search for all tweets with the hashtag “#FavoriteIceCream” in order to find other tweets containing (and presumably related to) that hashtag.

[0096] Typical speech-to-text systems are not suitable, however, for composing messages that include properly formatted TWITTER usernames and hashtags. For example, typical speech-to-text systems attempt to place spaces between each successive word uttered by a user, which makes it difficult to input a hashtag or username that is composed of more than one word strung together without spaces. Thus, if a user intended to input the hashtag “#FavoriteIceCream” using a typical speech-to-text system, the user’s utterance would be transcribed as separate words, resulting in the text “hashtag favorite ice cream,” which fails to capture the user’s intent. Indeed, a typical speech-to-text system will not recognize that “hashtag” indicates the beginning of a special textual identifier, nor will it identify that one or more of the uttered words should be concatenated into one continuous string of characters. This problem is exacerbated by the fact that any user may create new hashtags at any time: there is no rule by which the typical speech-to-text system could identify which words following the word “hashtag” (or some other suitable keyword) should be concatenated. This is especially true because hashtags need not be placed at the end of a message—they may be placed at the beginning or in the middle of a message. For example, a user may intend to compose a message saying “Trying my #FavoriteIceCream at the ice cream stand. Rocky Road!” Typical speech-to-text transcriptions may transcribe the utterance as “hashtag

favorite ice cream at the ice cream stand...” or “#favorite ice cream at the ice cream stand...” neither of which captures the user’s intent. Usernames present similar problems, because they may be composed of multiple words, letters, numbers, etc. For example, a username “@HannahBee2” would be difficult to correctly transcribe using traditional speech-to-text systems.

[0097] Accordingly, it would be beneficial to provide a digital assistant that can correctly transcribe usernames and hashtags (and/or other special textual identifiers) for inclusion into messages, tweets, emails, and the like. In some embodiments, a digital assistant (e.g., digital assistant system 300) is configured to recognize that certain parts of a transcribed speech input correspond to a special textual identifier such as a username or a hashtag. In some embodiments, the digital assistant is configured to recognize one or more keywords indicating that one or more words, letters, numbers, and/or symbols following or preceding the keyword should be replaced with a properly formatted username or hashtag. In some embodiments, the digital assistant determines whether one or more of the words, letters, etc., following or preceding a keyword correspond to a known hashtag or username. The following discussion provides details of a digital assistant that can recognize special textual identifiers as described.

[0098] Figure 4A illustrates an example of a user interface 400 showing a possible dialogue between a user and a digital assistant in accordance with some embodiments. In this example, the digital assistant module 326 (e.g., using the speech to text processing module 330) transcribed an utterance from a user including the words “Tweet having lunch with at Hannah Baxter.” The digital assistant 326 uses natural language processing (e.g., with the natural language processing module 332) to determine that the user is requesting the digital assistant to post a message to the user’s TWITTER feed.

[0099] In some embodiments, the digital assistant displays this transcription to the user in bubble 402. In this example, the words “at Hannah Baxter” suggest that the user intended to include Hannah Baxter’s username in the tweet. However, the username is not properly reflected in the transcription. Thus, the digital assistant identifies that “at Hannah Baxter” should be replaced with a username associated with Hannah Baxter.

[00100] In some embodiments, in order to identify and replace text with an intended username, the digital assistant 326 recognizes a particular word in the transcribed utterance as a keyword indicating that one or more of the following words (or nearby words) refer to a

particular person who is associated with a username. In this example, the digital assistant 326 recognizes the word “at” as the keyword, though other keywords may also be used. In some cases, a user will simply recite a person’s name (e.g., “Hannah Baxter”), and the digital assistant will determine the appropriate username to include (e.g., “@HannahBee2”). In other cases, a user will recite an actual username (e.g., by saying “at Hannah Be Two”).

[00101] Returning to Figure 4A, the digital assistant identifies the appropriate username, and replaces the transcription “at Hannah Baxter” with the username “@HannahBee2”. In some embodiments, the digital assistant identifies usernames included in a tweet by searching a contact list associated with the user of the digital assistant. In the present example, the digital assistant may search a contact list on a user device 104 in order to determine the appropriate username of a person named Hannah Baxter. In some embodiments, the digital assistant 326 identifies usernames by searching a list of usernames within a social network, such as all usernames within the social network, or usernames of people that are specifically associated with the user (e.g., usernames of the user’s followers, usernames of people that the user is following, etc.). Once the username is identified, the digital assistant 326 may present the tweet, including the properly formatted username, in the preview bubble 404. In some embodiments, the preview bubble 404 is displayed without also displaying the bubble 402 with the first transcribed input.

[00102] As noted above, a user may dictate an actual username, rather than simply referring to the proper name of a particular user. Figure 4B illustrates an example of another user interface 410 showing a possible dialogue between a user and a digital assistant in accordance with some embodiments. In this example, the user dictated a verbal representation of the username associated with Hannah Baxter. Bubble 412 displays a transcription of the user’s utterance: “Tweet having lunch with at Hannah Be Two.” The digital assistant 326 can recognize that one or more words following “at” might represent a username, and search for possible usernames with which to replace the transcribed text. For example, the digital assistant may search a contact list on the user device 104 (or a group of usernames registered within the social network) for users that have a username corresponding to “@Hannah,” “@HannahB,” “@HannahBee,” “@HannahB2,” etc. If the digital assistant identifies a username that is an appropriate match, the transcribed text can be replaced with the identified username. If the digital assistant identifies more than one possible match (e.g., “@HannahBToo” and “@HannahBee2”), the digital assistant may engage in a dialogue with the user to disambiguate the input. The bubble 414 illustrates a possible disambiguation

dialogue, where the digital assistant 326 asks the user to specify which username was intended. The user may select the intended username, via voice or touch input, for example. The selected username is then included in the preview bubble 416 for the user to review. In some embodiments, if no username can be identified, the digital assistant will not replace the transcribed words with a username, and will simply use the transcribed words as identified by the speech-to-text processing module 330 and/or the natural language processing module 332.

[00103] In some embodiments, a digital assistant 326 does not rely on keywords in order to automatically identify and replace input text with usernames or other textual identifiers. For example, a user may provide an utterance that includes a name of a person without any keywords, such as “Tweet having lunch with Hannah Baxter.” In some embodiments, the digital assistant will search the input text for words that correspond to names in the user’s contact list, or names of people that are associated with any of the user’s social networks. If a name is found in the input text, the digital assistant may replace the name with the appropriate username (e.g., a TWITTER username). In some embodiments, the digital assistant will prompt the user to confirm whether an identified username should be used in place of the plain-text name.

[00104] Another example of a textual identifier that a digital assistant 326 can recognize in a spoken utterance is a hashtag. As described above with reference to usernames, in some embodiments, the digital assistant is configured to replace one or more words following a certain keyword with a properly formatted hashtag. A properly formatted hashtag may comprise a string of words, letters, numbers, and/or symbols that follow a “#” symbol (without interstitial spaces).

[00105] Figure 5A illustrates an example of a user interface 500 showing a possible dialogue between a user and a digital assistant in accordance with some embodiments. In this example, the digital assistant 326 (e.g., using the speech to text processing module 330) transcribed an utterance from a user including the words “Tweet I don’t care what flavor, just give me lots hashtag favorite ice cream.” In some embodiments, the digital assistant 326 provides this transcription in the bubble 502. Figure 5A illustrates an example where the term “hashtag” acts as a keyword to indicate that one or more of the following (or nearby) words should be replaced with a properly formatted hashtag. Other keywords could be used instead or in addition to “hashtag,” including, for example, “hash,” “tag,” “pound,” “mark,” etc. Upon detection of the keyword, the digital assistant 326 will attempt to replace one or

more of the words following (or preceding) the keyword with a properly formatted hashtag. In some embodiments, the digital assistant creates a properly formatted hashtag by concatenating all (or at least some) of the words following the keyword, and adding the proper symbol to the beginning of the concatenated text. As shown in the preview bubble 504, the digital assistant 326 has replaced the text “hashtag favorite ice cream” with “#FavoriteIceCream.”

[00106] However, it is not always the case that every word following a keyword is intended to be part of the hashtag. For example, a hashtag comprising multiple concatenated words may be placed at the beginning or middle of a message, making it difficult to determine which words should be converted to a hashtag, and which should not. Figure 5B illustrates an example of a user interface 510 showing a possible dialogue between a user and a digital assistant where the user placed a hashtag in between other textual portions of a message, in accordance with some embodiments. In this example, the digital assistant (e.g., using the speech to text processing module 330) transcribed an utterance from a user including the words “Tweet Vanilla. Can you believe my hashtag favorite ice cream is so boring?” In some embodiments, the digital assistant 326 provides this transcription in the bubble 512. Once again, the digital assistant 326 recognizes that the word “hashtag” indicates the possibility that the user intends one or more of the following words to be replaced with a properly formatted hashtag. In this case, however, it is not clear what those words may be. For example, it not clear whether the user intended the message to include “#Favorite,” “#FavoriteIceCream,” or “#FavoriteIceCreamIsSoBoring.” The problem is exacerbated by the fact that hashtags need not follow any particular grammar, or even be composed of complete words (e.g., #AMA, #M5, #CNN, etc.).

[00107] In some embodiments, in order to correctly identify text that is intended to correspond to a hashtag, the digital assistant 326 determines whether the user’s input corresponds to any of a set of known hashtags. Because hashtags often represent a popular social themes, ideas, sentiments, etc., people will include a hashtag in a message after seeing it in other people’s messages, or after seeing it in a list of popular hashtags. Thus, in some embodiments, the digital assistant 326 (in some embodiments, in combination with social networking service 122-6) searches among a set of currently popular hashtags for any that match one or more of the words following the keyword. In another example, the digital assistant 326 may search among a set of hashtags that are found in messages (e.g., tweets) that the user has recently read, or which are posted to the user’s account. The digital assistant

326 then attempts to identify popular hashtags that may match the words input by the user. In Figure 5B, the candidate words that may comprise the hashtag are those that follow the keyword: “favorite ice cream is so boring.” In some embodiments, the digital assistant 326 will search for various potential concatenations of these words (e.g., “#Favorite,” “#FavoriteIce,” “#FavoriteIceCream,” etc.) to determine a best candidate hashtag. If one of the hashtags matches (or is similar to), for example, a popular hashtag or a recently read hashtag, that particular combination of words and/or characters is converted into a properly formatted hashtag for inclusion into the user’s message. In some embodiments, matching algorithms also account for possible transcription errors in searching for candidate hashtags. For example, if a user’s utterance was incorrectly transcribed as “hashtag favorite eyes cream,” the digital assistant 326 can determine, using an appropriate algorithm (e.g., by searching previously used hashtags), that this is similar to “#FavoriteIceCream,” and replace the transcribed text accordingly.

[00108] In some embodiments, the digital assistant requests additional input from a user to disambiguate possible candidate hashtags. This may occur, for example, when the digital assistant finds multiple possible hashtags for different combinations of transcribed words, when there are two or more possible transcriptions, and/or when no popular hashtag is found and it is not obvious what words the user intended to be included in the hashtag. For example, the digital assistant may ask the user “Did you mean: #Favorite; #FavoriteIceCream; or #FavoriteIceCreamIsSoBoring?” The user may then select the intended hashtag, via voice or touch input, for example.

[00109] Once the digital assistant determines the best candidate hashtag, or once the user selects the intended hashtag, the formatted message may be displayed in the preview bubble 514, including the properly formatted hashtag “#FavoriteIceCream.”

[00110] In some embodiments, keywords can also be used to signal the end of an intended hashtag. Specifically, the digital assistant may recognize a first keyword indicating the beginning of a hashtag (e.g., the word “hashtag”), and a second keyword to indicate the end of a hashtag (e.g., “hashtag” or “end hashtag”). Other keywords may also be used instead of or in addition to these, such as “hash start” and “hash end,” or simply “hash” and “hash.” Figure 5C illustrates an example of a user interface 520 showing a dialogue between a user and a digital assistant where the user indicated both the beginning and the end of a hashtag with keywords, in accordance with some embodiments. In this example, the digital

assistant 326 (e.g., using the speech to text processing module 330) transcribed an utterance from a user including the words “Tweet Vanilla. Can you believe my hashtag favorite ice cream end hashtag is so boring?” In some embodiments, this transcription is provided in bubble 522. The digital assistant 326 recognizes the keywords “hashtag” and “end hashtag,” and creates a properly formatted hashtag using the words, numbers, and/or characters therebetween. As shown in preview bubble 524, the hashtag “#FavoriteIceCream” has been inserted into the message by the digital assistant 326 in response to recognizing the two keywords.

[00111] Figure 6 is a flow diagram illustrating a method 600 of recognizing textual identifiers within a plurality of words, in accordance with some embodiments. Textual identifiers are a type of special text that may be included in a body of text, and typically identify a particular person, theme, idea, comment, trend, etc. Textual identifiers include, for example, usernames (e.g., TWITTER usernames), tags (e.g., TWITTER hashtags), email addresses, and the like. In some embodiments, the method 600 is performed by the digital assistant system 300. In some embodiments, the method 600 is performed by the natural language processing module 332 of the digital assistant system 300.

[00112] In some embodiments, a voice input is received (601) from a user. The digital assistant receives (602) a textual representation of the user’s voice input, the textual representation including a plurality of words. In some embodiments, the textual representation is generated by the speech-to-text processing module 330. In some embodiments, the textual representation is a file or data structure containing text that represents of one or more words, characters, etc. In some embodiments, the voice input is received via a microphone (e.g., microphone 230) of the user device 104. In some embodiments, the voice input is received at the same device that receives the textual representation (e.g., the user device 104).

[00113] The digital assistant identifies (604) a keyword in the textual representation. In some embodiments, the keyword is used to indicate that one or more of the following or adjacent words comprise a textual identifier. Any word(s) may be chosen as a keyword (including characters, numbers, letters, etc.), and the present method may recognize multiple types of textual identifiers, each identified by one or more keywords. In some embodiments, the keyword signaling a username is “at.” In some embodiments, the keyword signaling a

hashtag is “hashtag.” In some embodiments, the keyword for a textual identifier corresponds to a symbol that signifies the textual identifier (e.g., “@” for a username, “#” for a hashtag).

[00114] In some embodiments, certain keywords are recognized only in the appropriate context. For example, if a user is composing a typical text message or email, the digital assistant may not attempt to identify usernames or hashtags, for example, by detecting keywords. In some embodiments, different keywords are used to trigger different behaviors in different contexts. For example, when composing a TWITTER message, the method may recognize the keywords “at” and/or “hashtag.” When composing a different type of message or posting, however, the method may recognize the keywords “nickname” and/or “topic.”

[00115] The digital assistant determines (606) whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers. In some embodiments, the one or more adjacent words precede the keyword. In some embodiments, the one or more adjacent words follow the keyword. In some embodiments, the one or more adjacent words include words following and words preceding the keyword.

[00116] In some embodiments, the collection of textual identifiers is a set of usernames in a contact list associated with a user. In some embodiments, the collection is a set of usernames previously input by the user. In some embodiments, the collection is a set of usernames associated with a social networking account of the user. In some embodiments, the collection includes some or all usernames that are registered in a social network. In some embodiments, the collection includes hashtags that were previously input by the user. In some embodiments, the collection includes hashtags that were previously identified by a social network. In some embodiments, the collection includes hashtags that meet a predefined popularity metric. An example of a popularity metric in accordance with some embodiments is a frequency of appearance of the hashtag in a social network within a predefined time period.

[00117] Responsive to a determination that the one or more adjacent words correspond to a textual identifier, the digital assistant replaces (608) the keyword and the one or more adjacent words with the textual identifier. In some embodiments, at least one of the one or more adjacent words are composed only of a single letter. In some embodiments, the textual identifier is a concatenation of the one or more adjacent words without interstitial spaces. In some embodiments, the textual identifier is a concatenation of a symbol and the one or more

adjacent words without interstitial spaces. In some embodiments, the symbol is “@.” In some embodiments, the symbol is “#.”

[00118] In some cases, a user may wish to include a keyword (e.g., “at,” “hashtag,” “tag,” etc.) in a message, but does not want the word to identify a textual identifier.

Accordingly, in some embodiments, responsive to determining that the one or more adjacent words do not correspond to a textual identifier, the digital assistant will not replace (610) the keyword with the word.

[00119] Figure 7 is a flow diagram illustrating a method 700 of recognizing textual identifiers within a plurality of words, in accordance with some embodiments. In some embodiments, the method 700 is performed by the natural language processing module 332 of the digital assistant system 300.

[00120] In some embodiments, a voice input is received (701) from a user. The digital assistant receives (702) a textual representation of the user’s voice input, the textual representation including a plurality of words. In some embodiments, the textual representation is generated by the speech-to-text processing module 330. In some embodiments, the textual representation is a file or data structure containing text that represents of one or more words, characters, etc. In some embodiments, the voice input is received via a microphone (e.g., microphone 230) of a user device 104. In some embodiments, the voice input is received at the same device that receives the textual representation (e.g., the user device 104).

[00121] The digital assistant identifies (704) a first keyword in the textual representation, the first keyword indicating the beginning of a textual tag. In some embodiments, the first keyword is “hashtag,” though any other suitable keyword could be used, such as “at,” “begin tag,” “tag,” etc.

[00122] The digital assistant identifies (706) a second keyword in the textual representation, the second keyword indicating the end of a textual tag. In some embodiments, the second keyword is “end hashtag,” though any other suitable keyword could be used, such as “hashtag,” “end tag,” “end at,” etc.

[00123] The digital assistant identifies (708) one or more words between the first keyword and the second keyword. The digital assistant then replaces (710) the first keyword, the second keyword, and the one or more words therebetween with a textual tag, wherein the textual tag comprises the one or more words. In some embodiments, the textual tag is a

hashtag. In some embodiments, the textual tag comprises a symbol (such as “#”) preceding the one or more words.

[00124] Figure 8 shows a functional block diagram of an electronic device 800 configured in accordance with the principles of the ideas described above. The functional blocks of the device may be implemented by hardware, software, or a combination of hardware and software to carry out the above described ideas. It is understood by persons of skill in the art that the functional blocks described in Figure 8 may be combined or separated into sub-blocks in various embodiments. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[00125] As shown in Figure 8, the electronic device 800 includes a text receiving unit 802 configured to receive a textual representation of a user's voice input, the textual representation including a plurality of words. The electronic device also includes a processing unit 804 coupled to the input receiving unit 802. In some embodiments, the processing unit 804 includes a keyword identifying unit 806, a textual identifier identifying unit 808, and a textual identifier replacement unit 810.

[00126] The processing unit 804 is configured to identify a keyword in the textual representation (e.g., with the keyword identifying unit 806). In some embodiments, the processing unit 804 is configured to identify first and second keywords in the textual representation (e.g., a keyword indicating the beginning of a textual identifier, and a keyword indicating the end of a textual identifier).

[00127] In some embodiments, the processing unit 804 is further configured to determine whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers (e.g., with the textual identifier identification unit 808). In some embodiments, the textual identifier is a hashtag. In some embodiments, the textual identifier is a username. The processor is further configured to, responsive to a determination that the one or more adjacent words correspond to a textual identifier, replace the keyword and the one or more adjacent words with the textual identifier (e.g., with the textual identifier replacement unit 810).

[00128] In some embodiments, the processing unit 804 is configured to identify first and second keywords in the textual representation (e.g., a keyword indicating the beginning of a textual identifier, and a keyword indicating the end of a textual identifier). In some

embodiments, the processing unit 804 is configured to identify one or more words between the first keyword and the second keyword (e.g., with the textual identifier identifying unit 808, and replace the first keyword, the second keyword, and the one or more words therebetween with a textual tag, wherein the textual tag comprises the one or more words (e.g., with the textual identifier replacement unit 810).

[00129] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the disclosed embodiments to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles and practical applications of the disclosed ideas, to thereby enable others skilled in the art to best utilize them with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method of recognizing textual identifiers within a plurality of words, comprising:
at an electronic device having one or more processors and memory storing one or more programs that when executed by the one or more processors cause the electronic device to perform the method:
receiving a textual representation of a user's voice input, the textual representation including a plurality of words;
identifying a keyword in the textual representation;
determining whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers; and
responsive to a determination that the one or more adjacent words correspond to a textual identifier, replacing the keyword and the one or more adjacent words with the textual identifier.
2. The method of claim 1, further comprising receiving the voice input from the user.
3. The method of any of claims 1-2, wherein the textual identifier is a hashtag.
4. The method of claim 3, wherein the hashtag is a TWITTER hashtag.
5. The method of any of claims 3-4 wherein the hashtag includes the symbol “#” followed by one or more words.
6. The method of any of claims 3-5, wherein the collection includes hashtags that were previously identified by a social network.
7. The method of any of claims 3-6, wherein the collection includes hashtags that meet a predefined popularity metric.
8. The method of claim 7, wherein the predefined popularity metric is a frequency of appearance in a social network within a predefined time period.
9. The method of any of claims 1-2, wherein the textual identifier is a username.
10. The method of claim 9, wherein the username is a TWITTER username.

11. The method of any of claims 9-10, wherein the collection includes usernames that are registered in a social network.
12. The method of claim 11, wherein the social network is TWITTER.
13. The method of any of claims 9-12, wherein the collection is a contact list of the user.
14. The method of any of claims 9-12, wherein the collection is a set of usernames previously input by the user.
15. The method of any of claims 1-2, wherein the textual identifier is an email address.
16. The method of any of claims 1-15, wherein the adjacent words immediately follow the keyword.
17. The method of any of claims 1-16, wherein at least one of the one or more adjacent words is composed only of a single letter.
18. The method of any of claims 1-16, wherein the textual identifier is a concatenation of the one or more adjacent words without interstitial spaces.
19. The method of any of claims 1-16, wherein the textual identifier is a concatenation of a symbol and the one or more adjacent words without interstitial spaces.
20. The method of any of claims 1-19, further comprising, responsive to a determination that the one or more adjacent words do not correspond to a textual identifier, not replacing the keyword and the one or more adjacent words.
21. The method of any of claims 9-20, wherein the username includes the symbol "@" followed by one or more words.
22. A method of recognizing textual identifiers within a plurality of words, comprising:
 - at an electronic device having one or more processors and memory storing one or more programs that when executed by the one or more processors cause the electronic device to perform the method:
 - receiving a textual representation of a user's voice input, the textual representation including a plurality of words;

identifying a first keyword in the textual representation, the first keyword indicating the beginning of a textual tag;

identifying a second keyword in the textual representation, the second keyword indicating the end of a textual tag;

identifying one or more words between the first keyword and the second keyword; and

replacing the first keyword, the second keyword, and the one or more words therebetween with a textual tag, wherein the textual tag comprises the one or more words.

23. The method of claim 22, further comprising receiving the voice input from the user.

24. The method of any of claims 22-23, wherein the textual tag further comprises a symbol preceding the one or more words.

25. The method of claim 24, wherein the symbol is “#.”

26. The method of any of claims 22-25, wherein the first keyword is “hashtag,” and wherein the second keyword is “end hashtag.”

27. The method of any of claims 22-26, wherein the textual identifier is a concatenation of the one or more adjacent words without interstitial spaces.

28. An electronic device, comprising:

one or more processors;

memory; and

one or more programs, wherein the one or more programs are stored in the memory and configured to be executed by the one or more processors, the one or more

programs including instructions for:

receiving a textual representation of a user's voice input, the textual representation including a plurality of words;

identifying a keyword in the textual representation;

determining whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers; and

responsive to a determination that the one or more adjacent words correspond to a textual identifier, replacing the keyword and the one or more adjacent words with the textual identifier.

29. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by an electronic device, cause the device to:

- receive a textual representation of a user's voice input, the textual representation including a plurality of words;
- identify a keyword in the textual representation;
- determine whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers; and
- responsive to a determination that the one or more adjacent words correspond to a textual identifier, replace the keyword and the one or more adjacent words with the textual identifier.

30. An electronic device, comprising:

- means for receiving a textual representation of a user's voice input, the textual representation including a plurality of words;
- means for identifying a keyword in the textual representation;
- means for determining whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers; and
- means for, responsive to a determination that the one or more adjacent words correspond to a textual identifier, replacing the keyword and the one or more adjacent words with the textual identifier.

31. An information processing apparatus for use in an electronic device, comprising:

- means for receiving a textual representation of a user's voice input, the textual representation including a plurality of words;
- means for identifying a keyword in the textual representation;
- means for determining whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers; and
- means for, responsive to a determination that the one or more adjacent words correspond to a textual identifier, replacing the keyword and the one or more adjacent words with the textual identifier.

32. A portable electronic device or a computer system, comprising:

- one or more processors; and

memory storing one or more programs for execution by the one or more processors, the one or more programs including instructions for performing any of the methods of claims 1-27.

33. A computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which, when executed by a portable electronic device or a computer system with one or more processors, cause the device to perform any of the methods of claims 1-27.

34. A portable electronic device or a computer system, comprising:
means for performing any of the methods of claims 1-27.

35. An information processing apparatus for use in a portable electronic device or a computer system, comprising:
means for performing any of the methods of claims 1-27.

36. A portable electronic device or a computer system, comprising:
a processing unit configured to perform any of the methods of claims 1-27.

37. An electronic device, comprising:
a text receiving unit configured to receive a textual representation of a user's voice input, the textual representation including a plurality of words; and
a processing unit coupled to the text receiving unit, the processing unit configured to:
identify a keyword in the textual representation;
determine whether one or more words adjacent to the keyword correspond to a textual identifier of a collection of textual identifiers; and
responsive to a determination that the one or more adjacent words correspond to a textual identifier, replace the keyword and the one or more adjacent words with the textual identifier.

38. The electronic device of claim 37, the processing unit further configured to receive the voice input from the user.

39. The electronic device of any of claims 37-38, wherein the textual identifier is a hashtag.

40. The electronic device of claim 39, wherein the hashtag is a TWITTER hashtag.

41. The electronic device of any of claims 39-40, wherein the hashtag includes the symbol “#” followed by one or more words.
42. The electronic device of any of claims 39-41, wherein the collection includes hashtags that were previously identified by a social network.
43. The electronic device of any of claim 39-42, wherein the collection includes hashtags that meet a predefined popularity metric.
44. The electronic device of claim 43, wherein the predefined popularity metric is a frequency of appearance in a social network within a predefined time period.
45. The electronic device of any of claims 37-38, wherein the textual identifier is a username.
46. The electronic device of claim 45, wherein the username is a TWITTER username.
47. The electronic device of any of claims 45-46, wherein the collection includes usernames that are registered in a social network.
48. The electronic device of claim 47, wherein the social network is TWITTER.
49. The electronic device of any of claims 45-48, wherein the collection is a contact list of the user.
50. The electronic device of any of claims 45-48, wherein the collection is a set of usernames previously input by the user.
51. The electronic device of any of claims 37-38, wherein the textual identifier is an email address.
52. The electronic device of any of claims 37-51, wherein the adjacent words immediately follow the keyword.
53. The electronic device of any of claims 37-52, wherein at least one of the one or more adjacent words is composed only of a single letter.
54. The electronic device of any of claims 37-53, wherein the textual identifier is a concatenation of the one or more adjacent words without interstitial spaces.

55. The electronic device of any of claims 37-54, wherein the textual identifier is a concatenation of a symbol and the one or more adjacent words without interstitial spaces.
56. The electronic device of any of claims 37-55, the processing unit further configured to, responsive to a determination that the one or more adjacent words do not correspond to a textual identifier, not replace the keyword and the one or more adjacent words.
57. The electronic device of any of claims 45-56, wherein the username includes the symbol "@" followed by one or more words.
58. An electronic device, comprising:
a text receiving unit configured to receive a textual representation of a user's voice input, the textual representation including a plurality of words; and
a processing unit coupled to the text receiving unit, the processing unit configured to:
identify a first keyword in the textual representation, the first keyword indicating the beginning of a textual tag;
identify a second keyword in the textual representation, the second keyword indicating the end of a textual tag;
identify one or more words between the first keyword and the second keyword; and
replace the first keyword, the second keyword, and the one or more words therebetween with a textual tag, wherein the textual tag comprises the one or more words.
59. The electronic device of claim 58, further comprising receiving the voice input from the user.
60. The electronic device of any of claims 58-59, wherein the textual tag further comprises a symbol preceding the one or more words.
61. The electronic device of claim 60, wherein the symbol is "#."
62. The electronic device of any of claims 58-61, wherein the first keyword is "hashtag," and wherein the second keyword is "end hashtag."
63. The electronic device of any of claims 58-62, wherein the textual identifier is a concatenation of the one or more adjacent words without interstitial spaces.

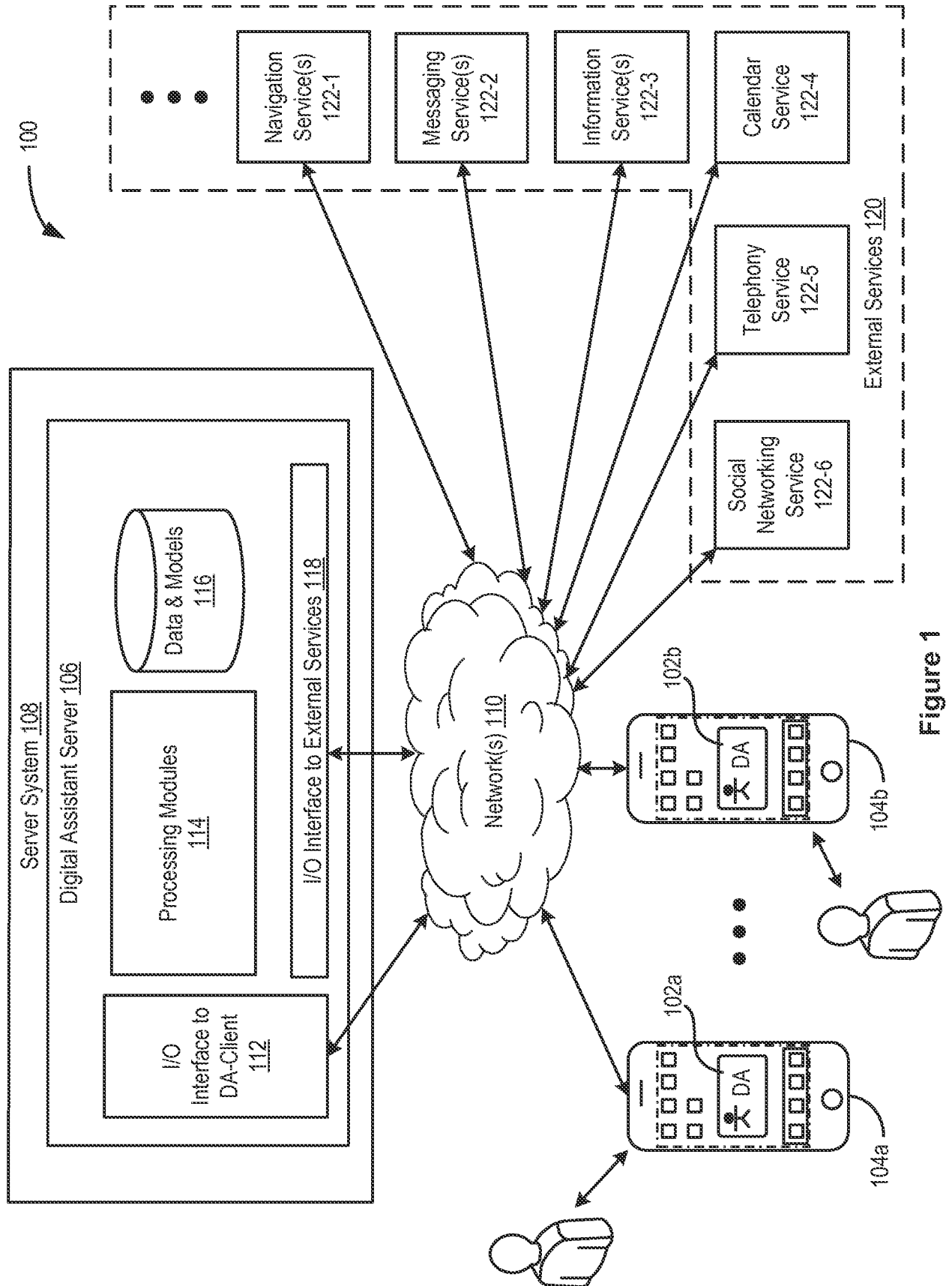


Figure 1

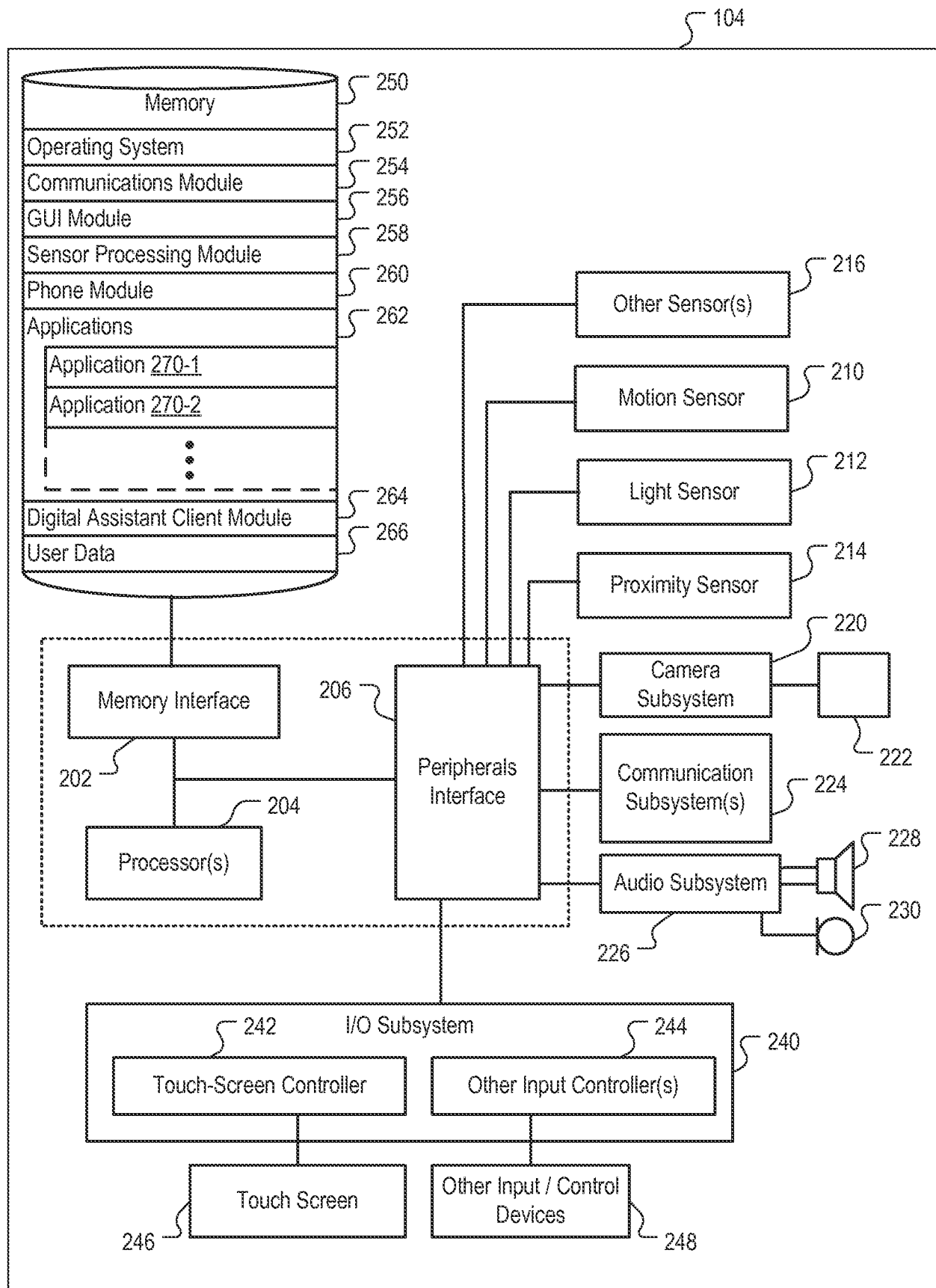


Figure 2

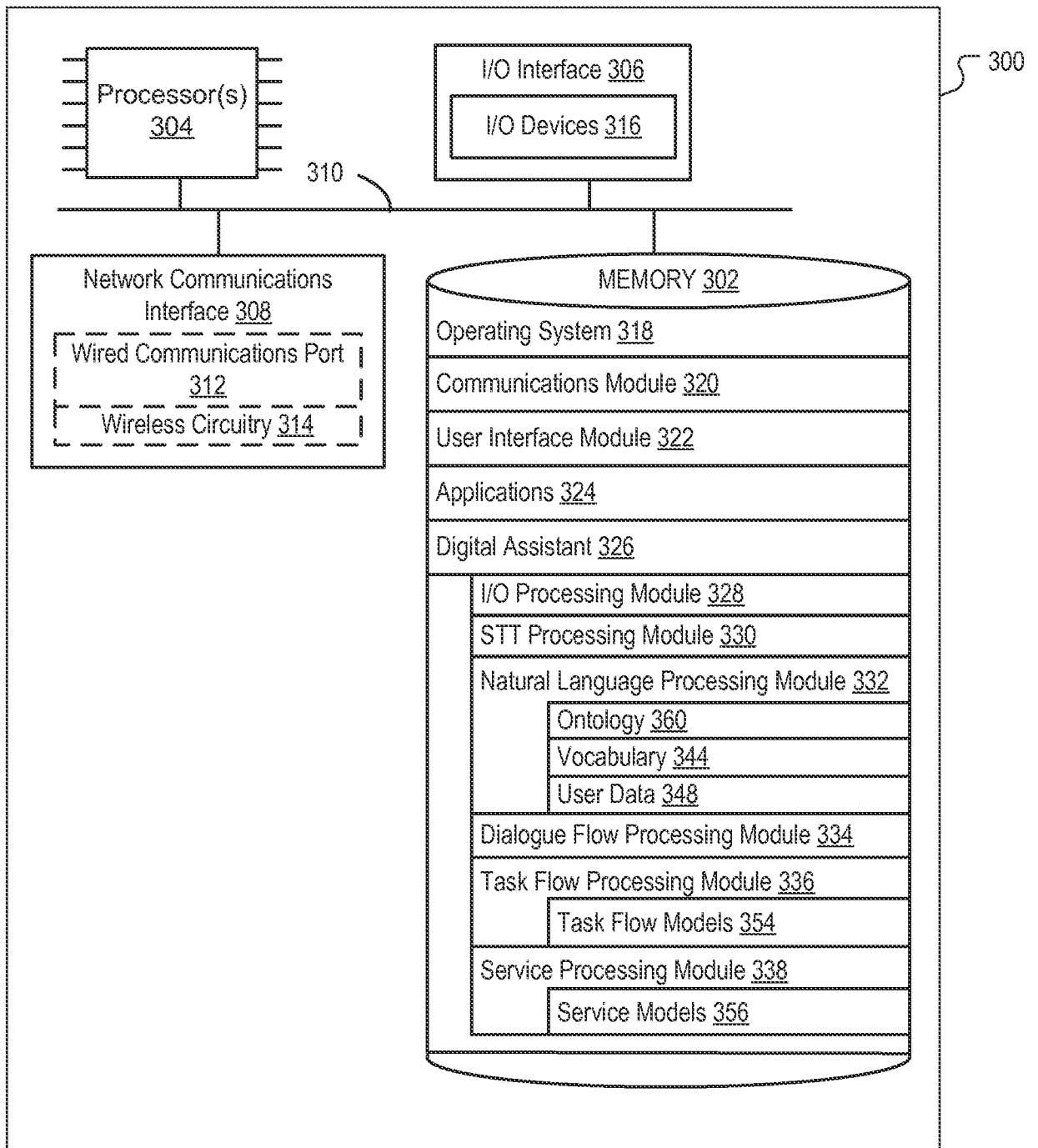


Figure 3A

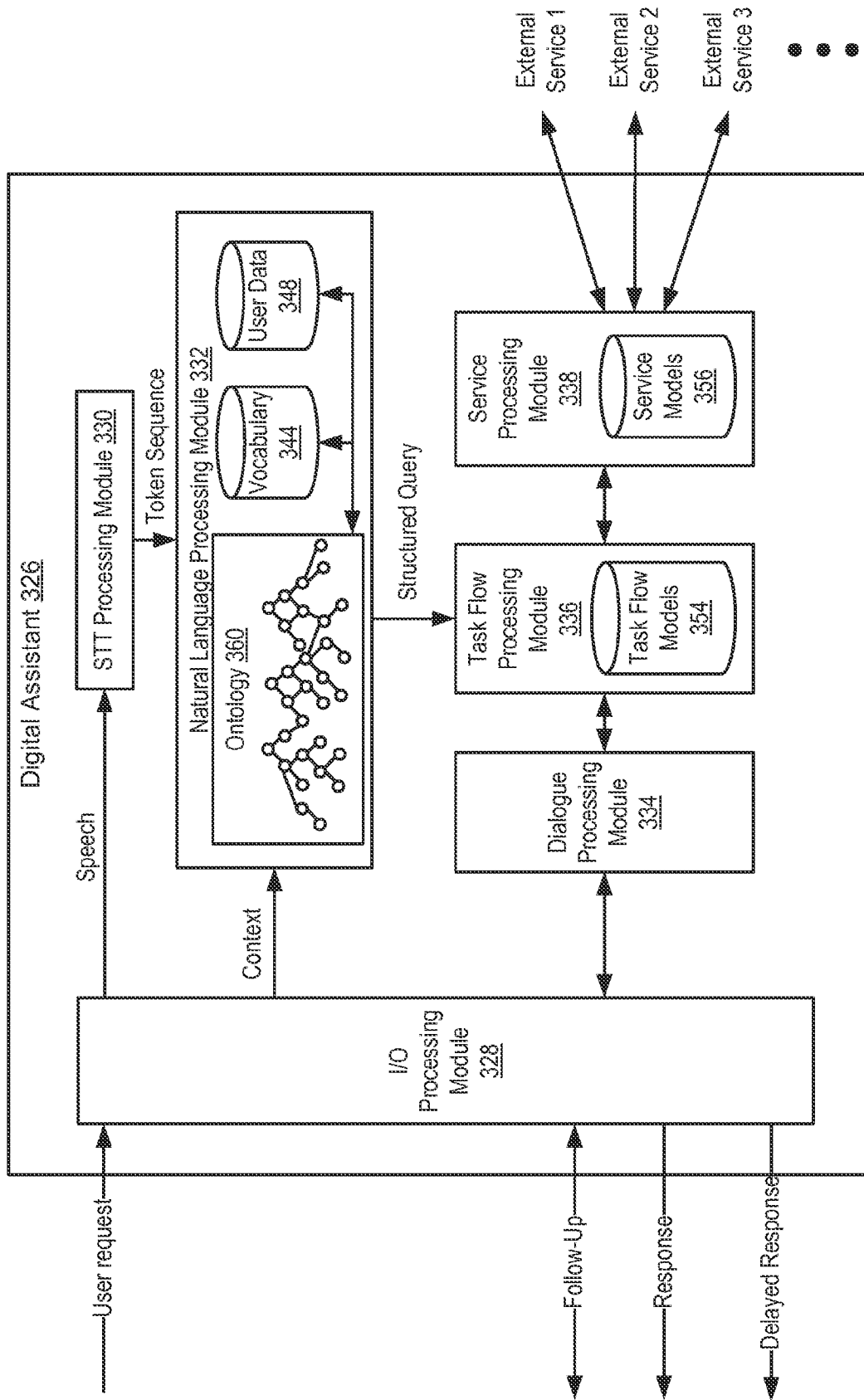


Figure 3B

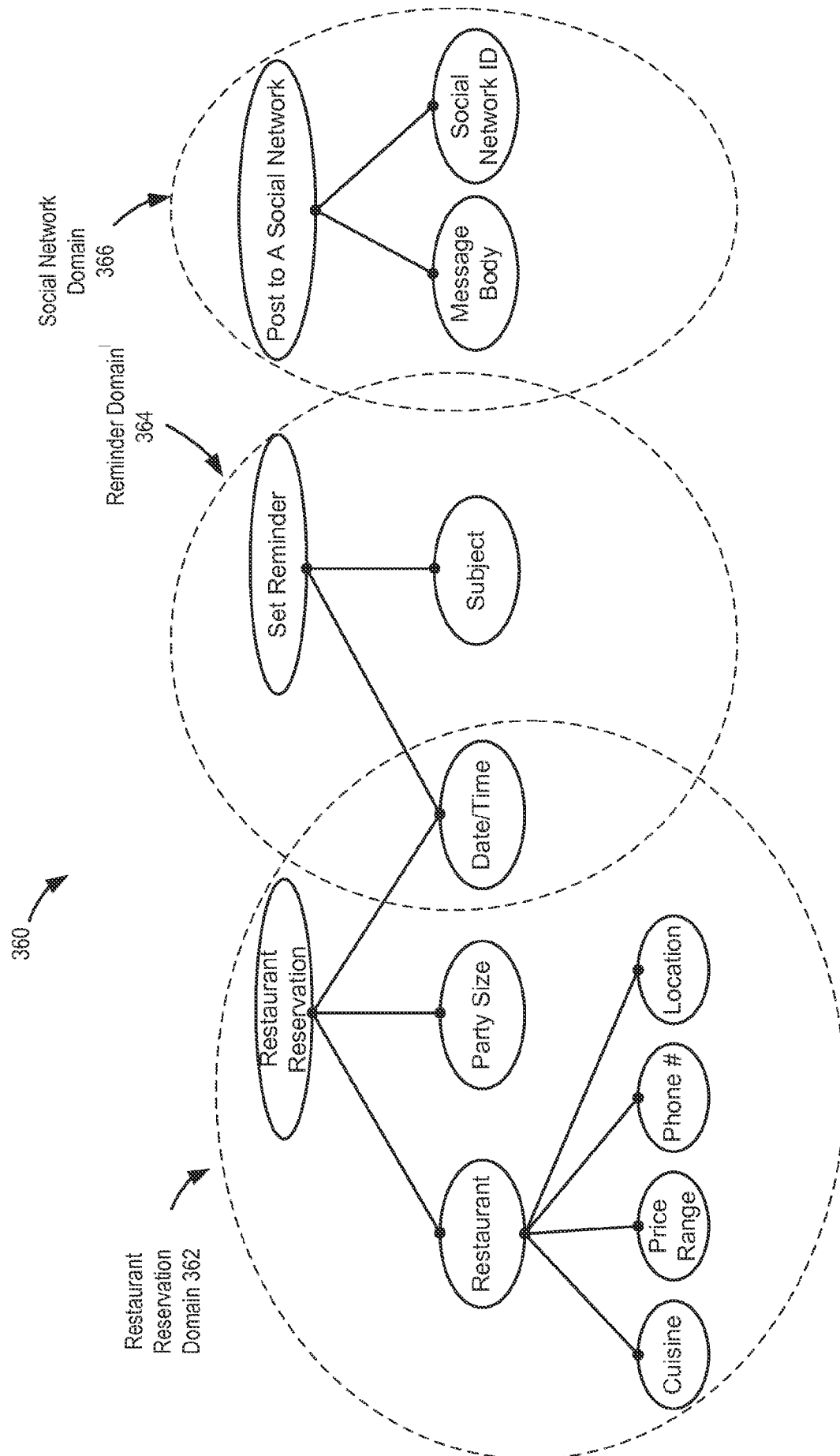


Figure 3C

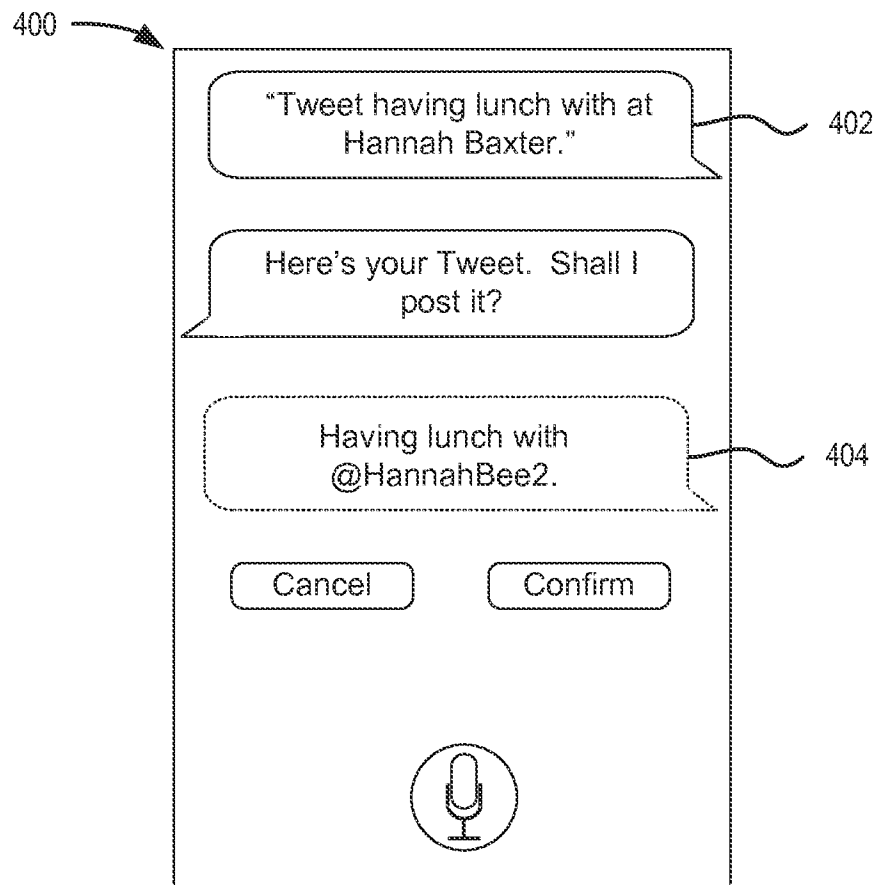


Figure 4A

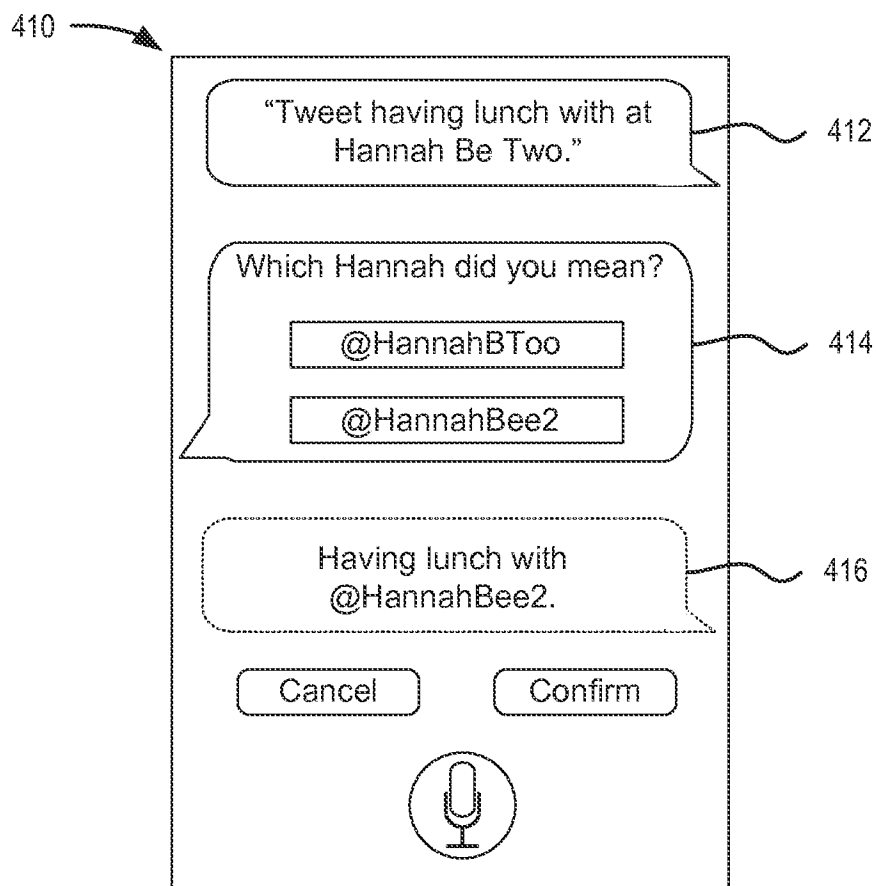


Figure 4B

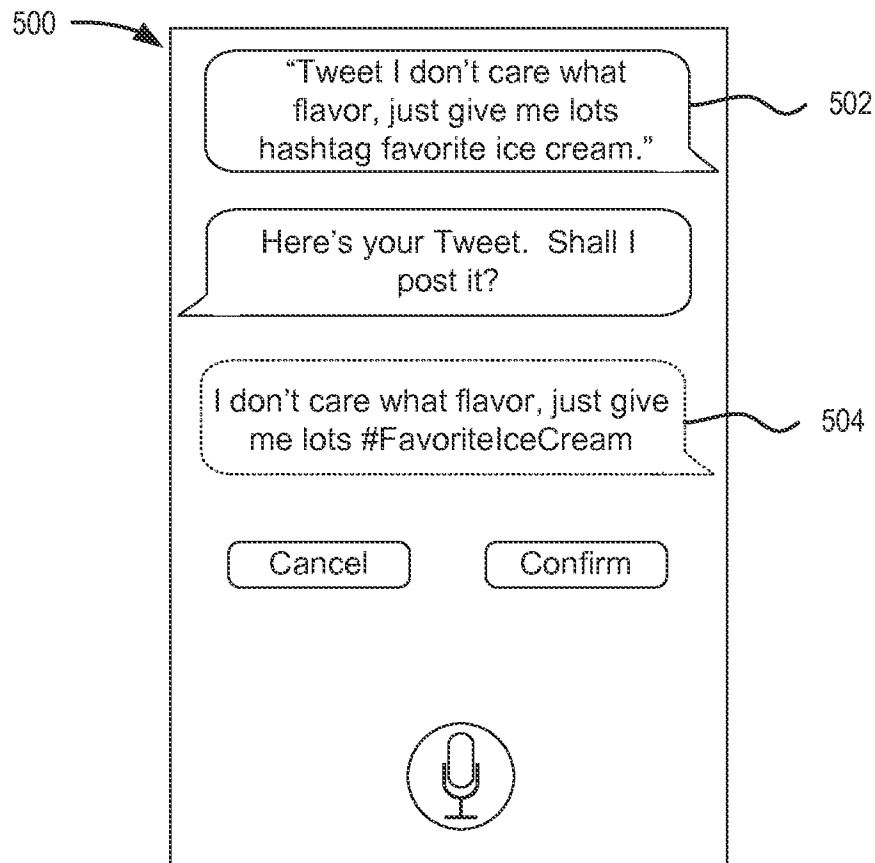


Figure 5A

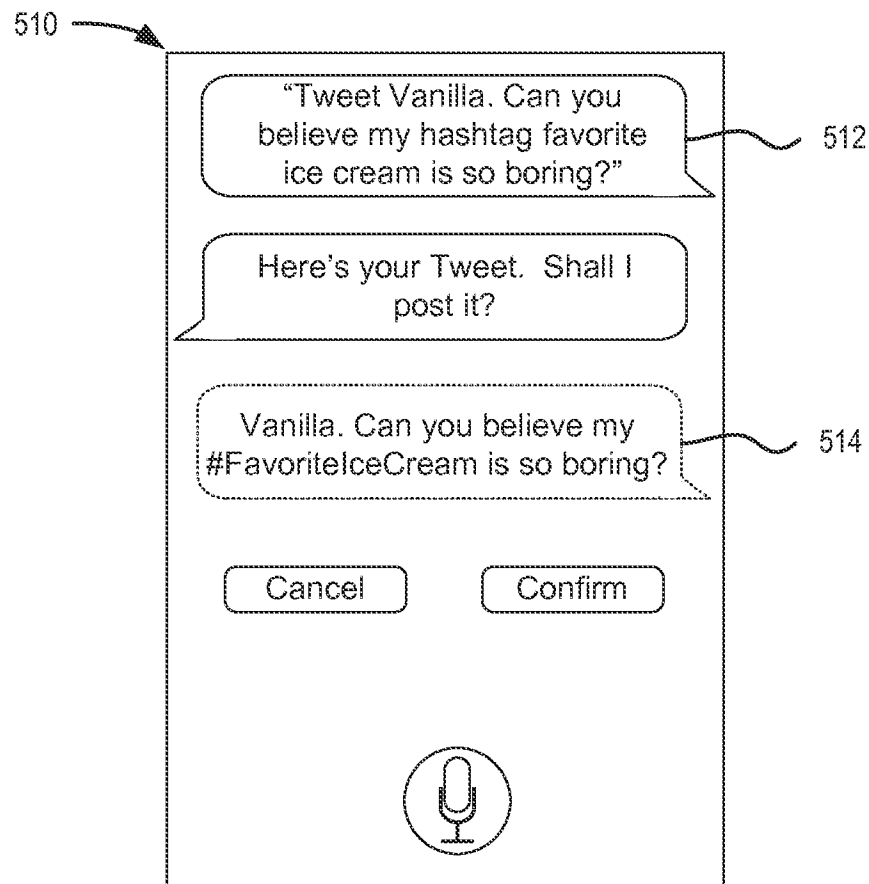


Figure 5B

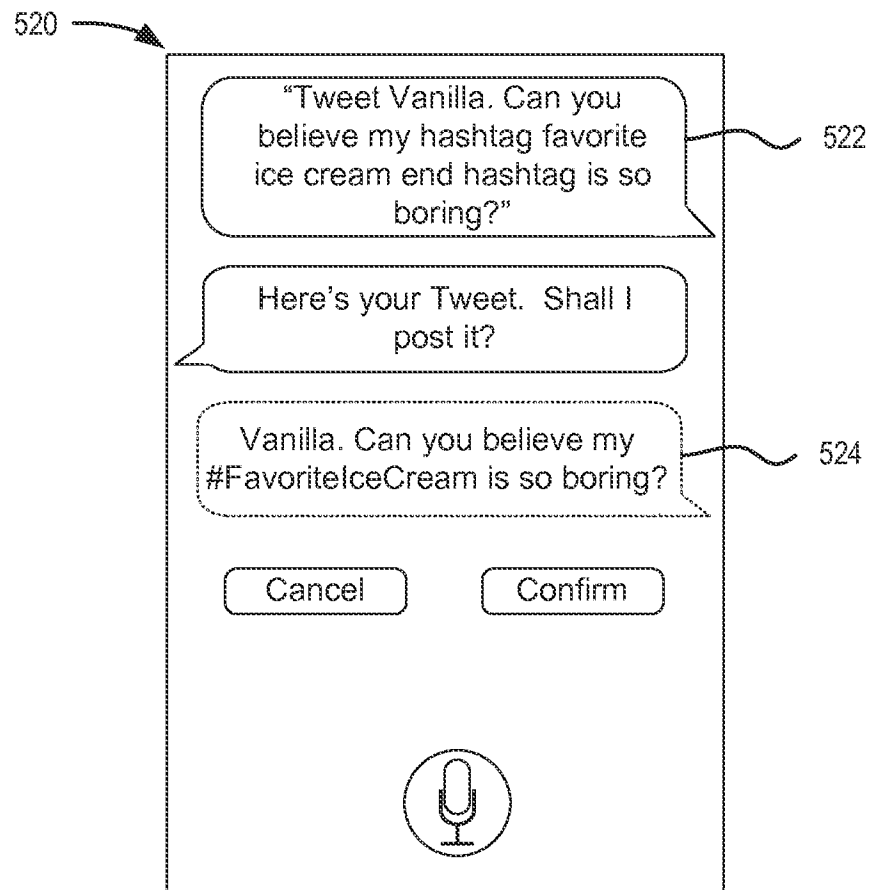


Figure 5C

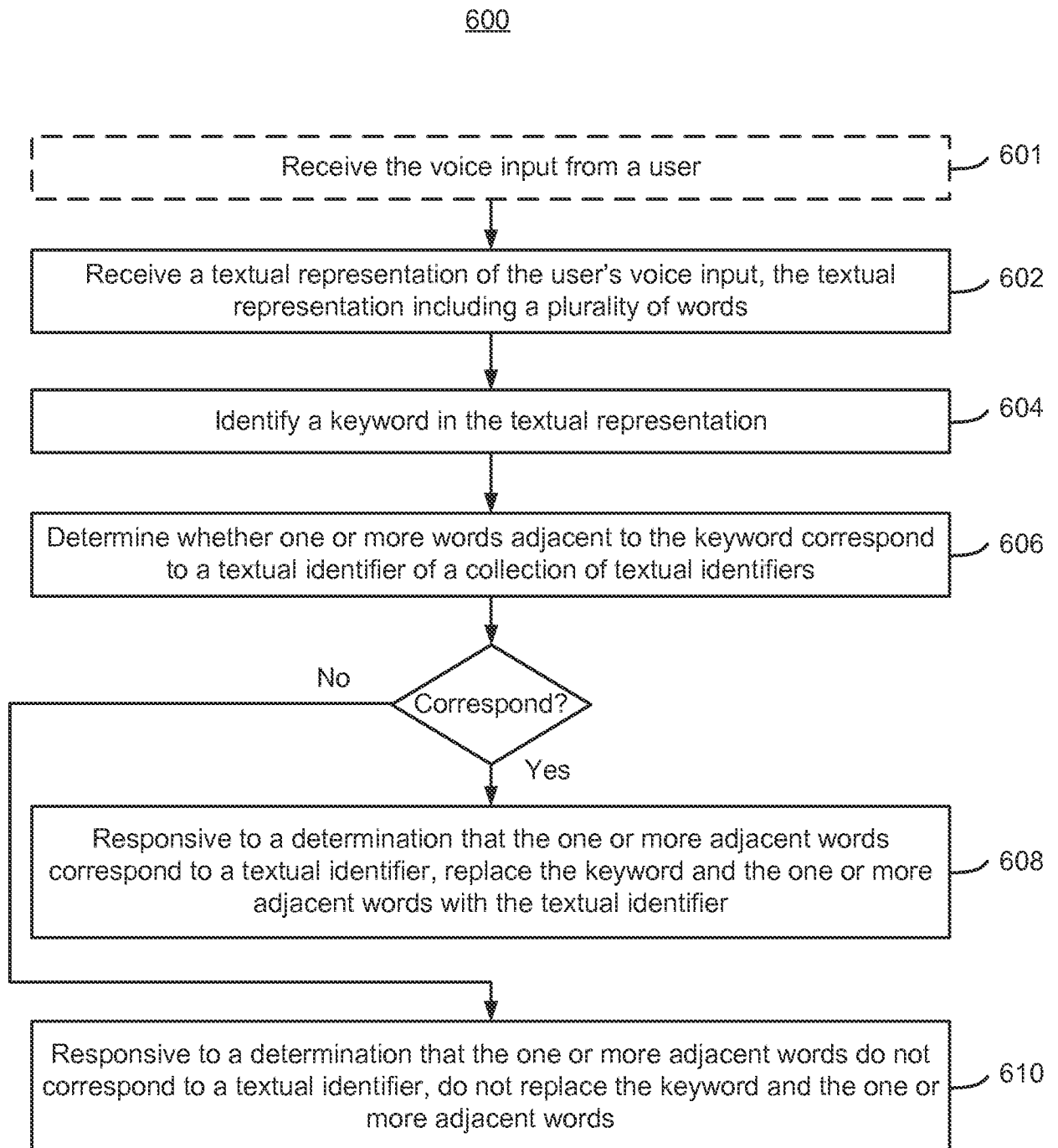
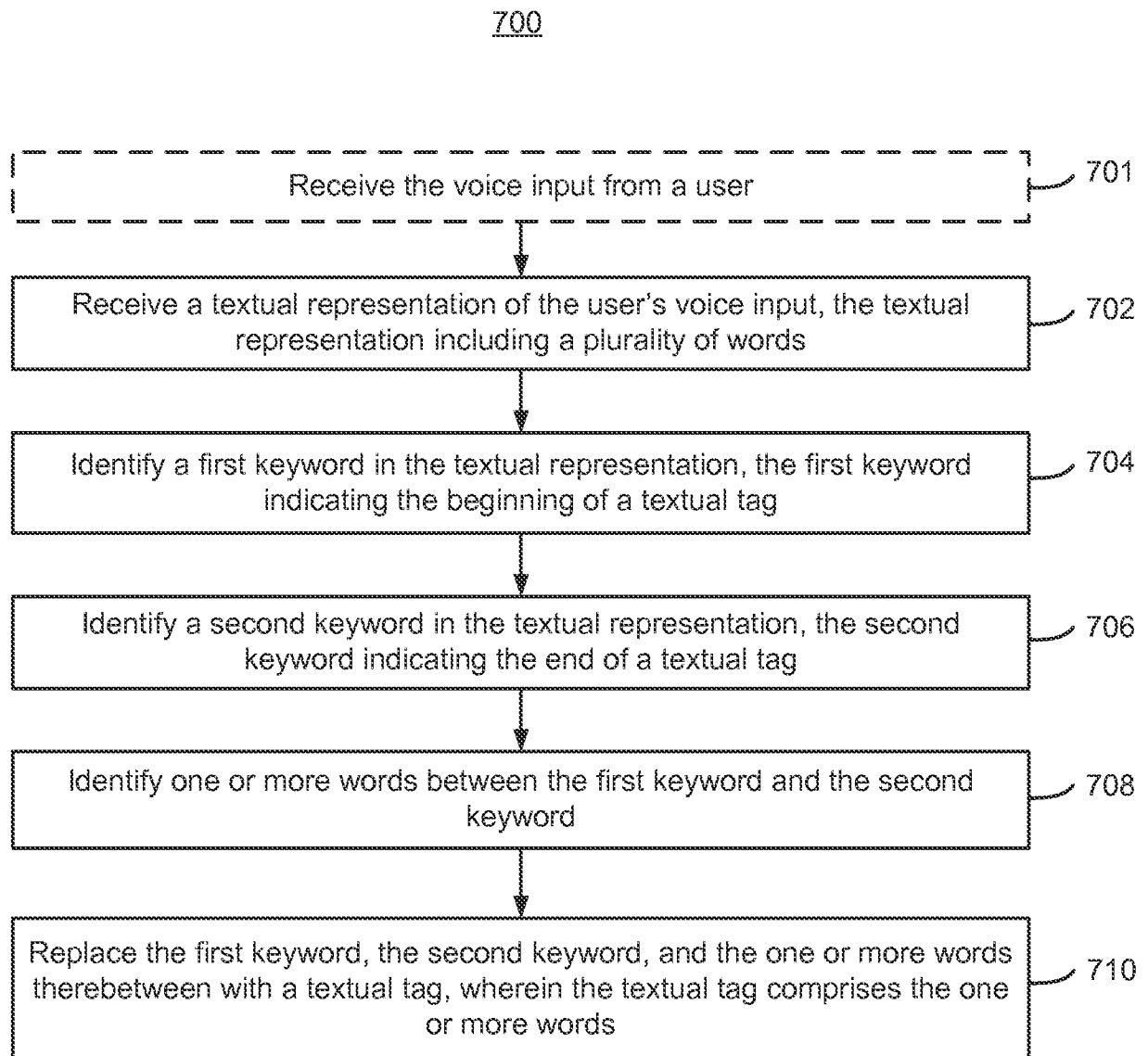


Figure 6

**Figure 7**

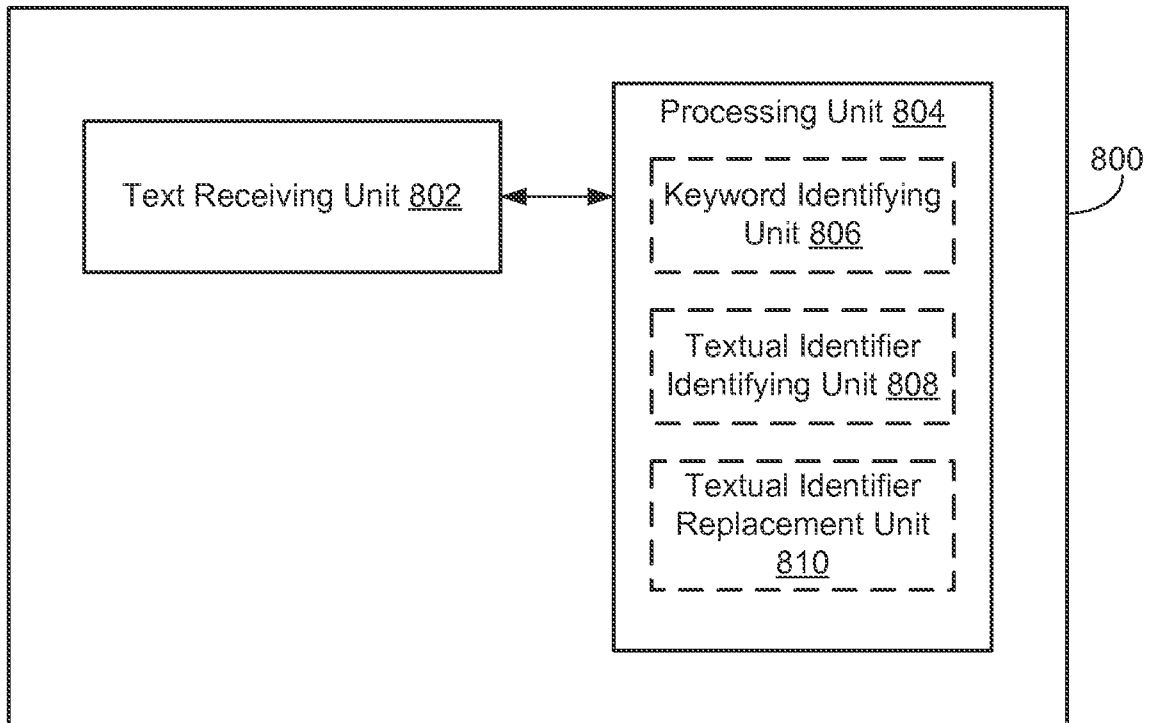


Figure 8