(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2018 (22.02.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/034928 A1

(51) International Patent Classification:
G06F 17/28 (2006.01) *G10L 15/22* (2006.01)
G06N 3/04 (2006.01) *G06Q 30/02* (2012.01)
G06N 5/02 (2006.01)

(21) International Application Number:
PCT/US2017/046243

(22) International Filing Date:
10 August 2017 (10.08.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/238,666 16 August 2016 (16.08.2016) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 15/238,666 (CON)
Filed on 16 August 2016 (16.08.2016)

(71) Applicant: **EBAY INC.** [US/US]; 2145 Hamilton Avenue, San Jose, California 95125 (US).

(72) Inventors: **GASKILL, Braddock**; 1720 Mission St., Apt 7, South Pasadena, California 91030 (US). **HAVIV, Adi Guila**; 310W 80 Street, New York, New York 10024 (US).

(74) Agent: **SCHEER, Bradley** et al.; Schwegman, Lundberg & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, SA, 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, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, 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).

(54) Title: SELECTING NEXT USER PROMPT TYPES

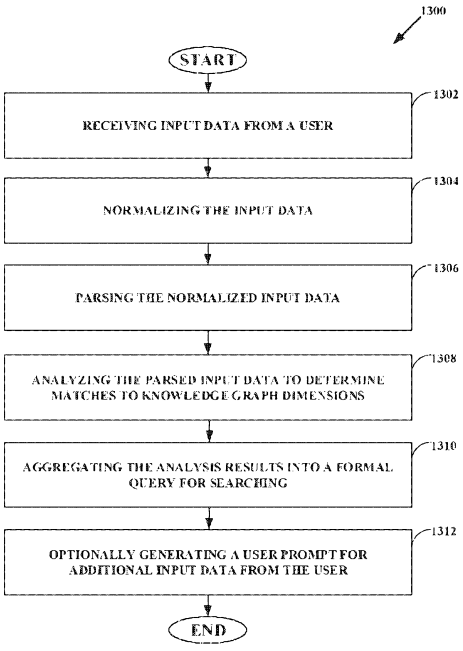

FIG. 13

(57) Abstract: Selecting types of generated prompts for further data from a user in a multi-turn interactive dialog. In one scenario, a processed sequence of user inputs and machine-generated prompts improves searches for the most relevant items available for purchase in an electronic marketplace. The number of prompts may be limited to a predetermined maximum value. Prompt generation is minimized by incorporating into a knowledge graph world knowledge that helps user intent inference. Prompt generation may be suppressed if a search indicates the reply to a prompt will not lead to any satisfactory search results. Prompts can provide suggestions for available search results that either meet all query constraints, or meet only some query constraints if a search indicates no search results are available that meet all query constraints. Prompts can provide suggested incisive reply phrasing likely to improve search results through an affirmation or negation reply.

WO 2018/034928 A1

[Continued on next page]

Published:
— *with international search report (Art. 21(3))*

SELECTING NEXT USER PROMPT TYPES

RELATED APPLICATIONS

[0001] This international application claims the benefit of priority to U.S. Patent Application Serial No. 15/238,666, filed on August 16, 2016, entitled "Selecting Next User Prompt Types In An Intelligent Online Personal Assistant Multi-Turn Dialog," which is incorporated herein by reference in its entirety.

BACKGROUND

[0002] Traditional searching is impersonal. One cannot speak to a traditional browsing engine in normal language. Conventional searching is time consuming, there is too much selection and much time can be wasted browsing pages of results. Trapped by the technical limitations of conventional tools, it is difficult for a user to communicate intent, for example a user cannot share photos of products to help with a search. As selection balloons to billions of items online, comparison searching has become more important than ever, while current solutions were not designed for this scale. Irrelevant results are often shown and do not bring out the best results. Traditional forms of comparison searching (search+refinements+browse) are no longer useful.

BRIEF SUMMARY

[0003] In one example, an intelligent personal assistant system includes scalable artificial intelligence (AI) that permeates the fabric of existing messaging platforms to provide an intelligent online personal assistant (or "bot"). The system may leverage existing inventories and curated databases to provide intelligent, personalized answers in predictive turns of communication between a human user and an intelligent online personal assistant. One example of an intelligent personal assistant system includes a knowledge graph. Machine learning components may continuously identify and learn from user intents so that user identity and understanding is enhanced over time. The user experience thus provided is inspiring, intuitive, unique and may be focused on the usage and behavioral patterns of certain age groups, such as millennials, for example.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0004] The drawings illustrate generally, by way of example, but not by way of limitation, various embodiments discussed in the present document. In order more easily to identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

[0005] FIG. 1 shows a networked system, according to some example embodiments.

[0006] FIG. 2 shows a general architecture of an intelligent personal assistant system, according to some example embodiments.

[0007] FIGS. 3A and 3B show components of a speech recognition component, according to some example embodiments.

[0008] FIG. 4 shows a representative software architecture software architecture, which may be used in conjunction with various hardware architectures described herein.

[0009] FIG. 5 shows components of a machine, according to some example embodiments, able to read instructions from a machine-readable medium (e.g., a computer-readable storage medium) and perform any one or more of the methodologies discussed herein.

[0010] FIG. 6 shows an example environment into which an intelligent online personal assistant can be deployed, according to some example embodiments.

[0011] FIG. 7 shows an overview of the intelligent personal assistant system processing natural language user inputs to generate an item recommendation in an electronic marketplace, according to some example embodiments.

[0012] FIG. 8 shows a natural language understanding (NLU) component, its sub-components, and other components with which it interacts, according to some example embodiments.

[0013] FIG. 9 shows the results of various analyses, according to some example embodiments.

[0014] FIG. 10 shows a knowledge graph, according to some example embodiments.

[0015] FIGs. 11A and 11B show a concise knowledge graph with an item category, some item attributes, and some item attribute values, according to some example embodiments.

[0016] FIG. 12 shows an overview of the intelligent personal assistant system processing natural language user inputs to generate suggestive prompts, according to some example embodiments.

[0017] FIG. 13 shows a flowchart of a methodology for processing natural language user inputs to generate an item recommendation, according to some example embodiments.

DETAILED DESCRIPTION

[0018] "CARRIER SIGNAL" in this context refers to any intangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine, and includes digital or analog communications signals or other intangible medium to facilitate communication of such instructions. Instructions may be transmitted or received over the network using a transmission medium via a network interface device and using any one of a number of well-known transfer protocols.

[0019] "CLIENT DEVICE" in this context refers to any machine that interfaces to a communications network to obtain resources from one or more server systems or other client devices. A client device may be, but is not limited to, a mobile phone, desktop computer, laptop, portable digital assistants (PDAs), smart phones, tablets, ultra books, netbooks, laptops, multi-processor systems, microprocessor-based or programmable consumer electronics, game consoles, set-top boxes, or any other communication device that a user may use to access a network.

[0020] "COMMUNICATIONS NETWORK" in this context refers to one or more portions of a network that may be an ad hoc network, an intranet, an extranet, a virtual private network (VPN), a local area network (LAN), a wireless LAN (WLAN), a wide area network (WAN), a wireless WAN (WWAN), a metropolitan area network (MAN), the Internet, a portion of the Internet, a portion of the Public Switched Telephone Network (PSTN), a plain old telephone service (POTS) network, a cellular telephone network, a wireless network, a Wi-Fi® network, another type of network, or a combination of two or more such networks. For example, a network or a portion of a network may include a wireless or cellular network and the coupling may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or other type of cellular or wireless coupling. In this example, the coupling may implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long Term

Evolution (LTE) standard, others defined by various standard setting organizations, other long range protocols, or other data transfer technology.

[0021] "COMPONENT" in this context refers to a device, physical entity or logic having boundaries defined by function or subroutine calls, branch points, application program interfaces (APIs), or other technologies that provide for the partitioning or modularization of particular processing or control functions. Components may be combined via their interfaces with other components to carry out a machine process. A component may be a packaged functional hardware unit designed for use with other components and a part of a program that usually performs a particular function of related functions. Components may constitute either software components (e.g., code embodied on a machine-readable medium) or hardware components. A "hardware component" is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various example embodiments, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware components of a computer system (e.g., a processor or a group of processors) may be configured by software (e.g., an application or application portion) as a hardware component that operates to perform certain operations as described herein. A hardware component may also be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware component may include dedicated circuitry or logic that is permanently configured to perform certain operations. A hardware component may be a special-purpose processor, such as a Field-Programmable Gate Array (FPGA) or an Application Specific Integrated Circuit (ASIC). A hardware component may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware component may include software executed by a general-purpose processor or other programmable processor. Once configured by such software, hardware components become specific machines (or specific components of a machine) uniquely tailored to perform the configured functions and are no longer general-purpose processors. It will be appreciated that the decision to implement a hardware component mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software) may be driven by cost and time considerations. Accordingly, the phrase "hardware component"(or "hardware-implemented component") should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein. Considering

embodiments in which hardware components are temporarily configured (e.g., programmed), each of the hardware components need not be configured or instantiated at any one instance in time. For example, where a hardware component comprises a general-purpose processor configured by software to become a special-purpose processor, the general-purpose processor may be configured as respectively different special-purpose processors (e.g., comprising different hardware components) at different times. Software accordingly configures a particular processor or processors, for example, to constitute a particular hardware component at one instance of time and to constitute a different hardware component at a different instance of time. Hardware components can provide information to, and receive information from, other hardware components. Accordingly, the described hardware components may be regarded as being communicatively coupled. Where multiple hardware components exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) between or among two or more of the hardware components. In embodiments in which multiple hardware components are configured or instantiated at different times, communications between such hardware components may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware components have access. For example, one hardware component may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware component may then, at a later time, access the memory device to retrieve and process the stored output. Hardware components may also initiate communications with input or output devices, and can operate on a resource (e.g., a collection of information). The various operations of example methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented components that operate to perform one or more operations or functions described herein. As used herein, "processor-implemented component" refers to a hardware component implemented using one or more processors. Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented components. Moreover, the one or more processors may also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service" (SaaS). For example, at least some

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

[0022] "MACHINE-READABLE MEDIUM" in this context refers to a component, device or other tangible media able to store instructions and data temporarily or permanently and may include, but is not be limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, optical media, magnetic media, cache memory, other types of storage (e.g., Erasable Programmable Read-Only Memory (EEPROM)) and/or any suitable combination thereof. The term "machine-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store instructions. The term "machine-readable readable medium" will also be taken to include any medium, or combination of multiple media, that is capable of storing instructions (e.g., code) for execution by a machine, such that the instructions, when executed by one or more processors of the machine, cause the machine to perform any one or more of the methodologies described herein. Accordingly, a "machine-readable readable medium" refers to a single storage apparatus or device, as well as "cloud-based" storage systems or storage networks that include multiple storage apparatus or devices. The term "machine-readable storage medium" excludes signals per se. A machine readable medium includes a machine readable storage medium and a transmission medium or carrier signal.

[0023] "PROCESSOR" in this context refers to any circuit or virtual circuit (a physical circuit emulated by logic executing on an actual processor) that manipulates data values according to control signals (e.g., "commands", "op codes", "machine code", etc.) and which produces corresponding output signals that are applied to operate a machine. A processor may, for example, be a Central Processing Unit (CPU), a Reduced Instruction Set Computing (RISC) processor, a Complex Instruction Set Computing (CISC) processor, a Graphics Processing Unit (GPU), a Digital Signal Processor (DSP), an Application Specific

Integrated Circuit (ASIC), a Radio-Frequency Integrated Circuit (RFIC)or any combination thereof. A processor may further be a multi-core processor having two or more independent processors (sometimes referred to as "cores") that may execute instructions contemporaneously.

[0024] A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings that form a part of this document: Copyright 2016, eBay Inc, All Rights Reserved.

[0025] The description that follows includes systems, methods, techniques, instruction sequences, and computing machine program products that embody illustrative embodiments of the disclosure. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide an understanding of various embodiments of the inventive subject matter. It will be evident, however, to those skilled in the art, that embodiments of the inventive subject matter may be practiced without these specific details. In general, well-known instruction instances, protocols, structures, and techniques are not necessarily shown in detail.

[0026] With reference to Figure 1, an example embodiment of a high-level SaaS network architecture 100 is shown. A networked system 116 provides server-side functionality via a network 110 (e.g., the Internet or wide area network (WAN)) to a client device 108. A web client 102 and a programmatic client, in the example form of an application 104 are hosted and execute on the client device 108. The networked system 116 includes and application server 122, which in turn hosts an intelligent personal assistant system 106 that provides a number of functions and services to the application 104 that accesses the networked system 116. The application 104 also provides a number of interfaces described herein, which present output of the tracking and analysis operations to a user of the client device 108.

[0027] The client device 108 enables a user to access and interact with the networked system 116. For instance, the user provides input (e.g., touch screen input or alphanumeric input) to the client device 108, and the input is communicated to the networked system 116 via the network 110. In this instance, the networked system 116, in response to receiving

the input from the user, communicates information back to the client device 108 via the network 110 to be presented to the user.

[0028] An Application Program Interface (API) server 118 and a web server 120 are coupled to, and provide programmatic and web interfaces respectively, to the application server 122. The application server 122 hosts an intelligent personal assistant system 106, which includes components or applications. The application server 122 is, in turn, shown to be coupled to a database server 124 that facilitates access to information storage repositories (e.g., a database/cloud 126). In an example embodiment, the database/cloud 126 includes storage devices that store information accessed and generated by the intelligent personal assistant system 106.

[0029] Additionally, a third party application 114, executing on a third party server 112, is shown as having programmatic access to the networked system 116 via the programmatic interface provided by the Application Program Interface (API) server 118. For example, the third party application 114, using information retrieved from the networked system 116, may support one or more features or functions on a website hosted by the third party.

[0030] Turning now specifically to the applications hosted by the client device 108, the web client 102 may access the various systems (e.g., intelligent personal assistant system 106) via the web interface supported by the web server 120. Similarly, the application 104 (e.g., an "app") accesses the various services and functions provided by the intelligent personal assistant system 106 via the programmatic interface provided by the Application Program Interface (API) server 118. The application 104 may, for example, an "app" executing on a client device 108, such as an iOS or Android OS application to enable user to access and input data on the networked system 116 in an off-line manner, and to perform batch-mode communications between the programmatic client application 104 and the networked system networked system 116.

[0031] Further, while the SaaS network architecture 100 shown in Figure 1 employs a client-server architecture, the present inventive subject matter is of course not limited to such an architecture, and could equally well find application in a distributed, or peer-to-peer, architecture system, for example. The intelligent personal assistant system 106 could also be implemented as a standalone software program, which does not necessarily have networking capabilities.

[0032] Figure 2 is a block diagram showing the general architecture of an intelligent personal assistant system 106, according to some example embodiments. Specifically, the

intelligent personal assistant system 106 is shown to include a front end component 202 (FE) by which the intelligent personal assistant system 106 communicates (e.g., over the network 110) with other systems within the SaaS network architecture 100. The front end component 202 can communicate with the fabric of existing messaging systems. As used herein, the term messaging fabric refers to a collection of APIs and services that can power third party platforms such as Facebook messenger, Microsoft Cortana and other "bots". In one example, a messaging fabric can support an online commerce ecosystem that allows users to interact with commercial intent. Output of the front end component 202 can be rendered in a display of a client device, such as the client device 108 in Figure 1 as part of an interface with an intelligent personal assistant, or "bot".

[0033] The front end component 202 of the intelligent personal assistant system 106 is coupled to a back end component 204 for the front end (BFF) that operates to link the front end component 202 with an artificial intelligence framework 128. The artificial intelligence framework 128 may include several components as discussed below. The data exchanged between various components and the function of each component may vary to some extent, depending on the particular implementation.

[0034] In one example of an intelligent personal assistant system 106, an AI orchestrator 206 orchestrates communication between components inside and outside the artificial intelligence framework 128. Input modalities for the AI orchestrator 206 may be derived from a computer vision component 208, a speech recognition component 210, and a text normalization component which may form part of the speech recognition component 210, for example. The computer vision component 208 may identify objects and attributes from visual input (e.g., a photo). The speech recognition component 210 may convert audio signals (e.g., spoken utterances) into text. A text normalization component may operate to make input normalization, such as language normalization by rendering emoticons into text, for example. Other normalization is possible such as orthographic normalization, foreign language normalization, conversational text normalization, and so forth.

[0035] The artificial intelligence framework 128 further includes a natural language understanding or NLU component 214 that operates to extract user intent and various intent parameters. The NLU component 214 is described in further detail beginning with Figure 8.

[0036] The artificial intelligence framework 128 further includes a dialog manager 216 that operates to understand a "completeness of specificity" (for example of an input, such as a search query or utterance) and decide on a next action type and a related parameter (e.g.,

"search" or "request further information from user"). For convenience, all user inputs in this description may be referred to as "utterances", whether in text, voice, or image-related formats.

[0037] In one example, the dialog manager 216 operates in association with a context manager 218 and a Natural Language Generation (NLG) component 212. The context manager 218 manages the context and communication of a user with respect to the intelligent online personal assistant (or "bot") and the assistant's associated artificial intelligence. The context manager 218 retains a short term history of user interactions. A longer term history of user preferences may be retained in an identity service 222, described below. Data entries in one or both of these histories may include the relevant intent and all parameters and all related results of a given input, bot interaction, or turn of communication, for example. The NLG component 212 operates to compose a natural language utterance out of an AI message to present to a user interacting with the intelligent bot.

[0038] A search component 220 is also included within the artificial intelligence framework 128. The search component 220 may have front and back end units. The back end unit may operate to manage item or product inventory and provide functions of searching against the inventory, optimizing towards a specific tuple of user intent and intent parameters. The search component 220 is designed to serve several billion queries per day globally against very large high quality inventories. The search component 220 can accommodate text, or Artificial Intelligence (AI) encoded voice and image inputs, and identify relevant inventory items to users based on explicit and derived query intents.

[0039] An identity service 222 component operates to manage user profiles, for example explicit information in the form of user attributes, e.g., "name", "age", "gender", "geolocation", but also implicit information in forms such as "information distillates" such as "user interest", or "similar persona", and so forth. The artificial intelligence framework 128 may comprise part of or operate in association with, the identity service 222. The identity service 222 includes a set of policies, APIs, and services that elegantly centralizes all user information, helping the artificial intelligence framework 128 to have "intelligent" insights into user intent. The identity service 222 can protect online retailers and users from fraud or malicious use of private information.

[0040] The identity service 222 of the present disclosure provides many advantages. The identity service 222 is a single central repository containing user identity and profile data. It may continuously enrich the user profile with new insights and updates. It uses account

linking and identity federation to map relationships of a user with a company, household, other accounts (e.g., core account), as well as a user's social graph of people and relationships. The identity service 222 evolves a rich notification system that communicates all and only the information the user wants at the times and media they choose.

[0041] In one example, the identity service 222 concentrates on unifying as much user information as possible in a central clearinghouse for search, AI, merchandising, and machine learning models to maximize each component's capability to deliver insights to each user. A single central repository contains user identity and profile data in a meticulously detailed schema. In an onboarding phase, the identity service 222 primes a user profile and understanding by mandatory authentication in a bot application. Any public information available from the source of authentication (e.g., social media) may be loaded. In sideboarding phases, the identity service 222 may augment the profile with information about the user that is gathered from public sources, user behaviors, interactions, and the explicit set of purposes the user tells the AI (e.g., shopping missions, inspirations, preferences). As the user interacts with the artificial intelligence framework 128, the identity service 222 gathers and infers more about the user and stores the explicit data, derived information, and updates probabilities and estimations of other statistical inferences. Over time, in profile enrichment phases, the identity service 222 also mines behavioral data such as clicks, impressions, and browse activities for derived information such as tastes, preferences, and shopping verticals. In identity federation and account linking phases, when communicated or inferred, the identity service 222 updates the user's household, employer, groups, affiliations, social graph, and other accounts, including shared accounts.

[0042] The functionalities of the artificial intelligence framework 128 can be grouped into multiple parts, for example decisioning and context parts. In one example, the decisioning part includes operations by the AI orchestrator 206, the NLU component 214, the dialog manager 216, the NLG component 212, the computer vision component 208 and speech recognition component 210. The context part of the AI functionality relates to the parameters (implicit and explicit) around a user and the communicated intent (for example, towards a given inventory, or otherwise). In order to measure and improve AI quality over time, the artificial intelligence framework 128 may be trained using sample queries (e.g., a dev set) and tested on a different set of queries (e.g., an eval set), where both sets may be developed by human curation. Also, the artificial intelligence framework 128 may be trained on transaction and interaction flows defined by experienced curation specialists, or human tastemaker override rules 224. The flows and the logic encoded within the various

components of the artificial intelligence framework 128 define what follow-up utterance or presentation (e.g., question, result set) is made by the intelligent assistant based on an identified user intent.

[0043] Reference is made further above to example input modalities of the intelligent online personal assistant or bot in an intelligent personal assistant system 106. The intelligent personal assistant system 106 seeks to understand a user's intent (e.g., targeted search, compare, shop/browse, and so forth) and any mandatory parameters (e.g., product, product category, item, and so forth) and/or optional parameters (e.g., explicit information such as attributes of item/product, occasion, and so forth) as well as implicit information (e.g., geolocation, personal preferences, age, and gender, and so forth) and respond to the user with a well thought out or "intelligent" response. Explicit input modalities may include text, speech, and visual input and can be enriched with implicit knowledge of user (e.g., geolocation, previous browse history, and so forth). Output modalities can include text (such as speech, or natural language sentences, or product-relevant information, and images on the screen of a smart device, e.g., client device 108. Input modalities thus refer to the different ways users can communicate with the bot. Input modalities can also include keyboard or mouse navigation, touch-sensitive gestures, and so forth.

[0044] In relation to a modality for the computer vision component 208, a photograph can often represent what a user is looking for better than text. The user may not know what an item is called, or it may be hard or even impossible to use text for fine detailed information that only an expert may know, for example a complicated pattern in apparel or a certain style in furniture. Moreover, it is inconvenient to type complex text queries on mobile phones, and long text queries typically have poor recall. Thus, key functionalities of the computer vision component 208 may include object localization, object recognition, optical character recognition (OCR) and matching against inventory based on visual cues from an image or video. A bot enabled with computer vision is advantageous when running on a mobile device which has a built-in camera. Powerful deep neural networks can be used to enable computer vision applications.

[0045] In one example, the dialog manager 216 has as sub-components the context manager 218 and the NLG component 212. As mentioned above, the dialog manager 216 operates to understand the "completeness of specificity" and deciding on a next action type and parameter (e.g., "search" or "request further information from user"). The context manager 218 operates to manage the context and communication of a given user towards the bot and its AI. The context manager 218 comprises two parts: a long term history and a

short term memory. Each context manager entry may describe the relevant intent and all parameters and all related results. The context is towards the inventory, as well as towards other, future sources of knowledge. The NLG component 212 operates to compose a natural language utterance out of an AI message to present to a user interacting with the intelligent bot.

[0046] Fluent, natural, informative, and even entertaining dialog between man and machine is a difficult technical problem that has been studied for much of the past century, yet is still considered unsolved. However, recent developments in AI have produced useful dialog systems such as Siri™ and Alexa™.

[0047] In an ecommerce example of an intelligent bot, an initial very helpful element in seeking to solve this problem is to leverage enormous sets of e-commerce data. Some of this data may be retained in proprietary databases or in the cloud e.g., database/cloud 126. Statistics about this data may be communicated to dialog manager 216 from the search component 220 as context. The artificial intelligence framework 128 may act directly upon utterances from the user, which may be run through speech recognition component 210, then the NLU component 214, and then passed to context manager 218 as semi-parsed data. The NLG component 212 may thus help the dialog manager 216 generate human-like questions and responses in text or speech to the user. The context manager 218 maintains the coherency of multi-turn and long term discourse between the user and the artificial intelligence framework 128.

[0048] Discrimination may be recommended to poll a vast e-commerce dataset for only relevant, useful information. In one example, the artificial intelligence framework 128 uses results from the search component 220 and intelligence within the search component 220 to provide this information. This information may be combined with the history of interaction from the context manager 218. The artificial intelligence framework 128 then may decide on the next turn of dialog, e.g., whether it should be a question, or a "grounding statement" to validate, for example, an existing understanding or user intent, or an item recommendation (or, for example, any combination of all three). These decisions may be made by a combination of the dataset, the chat history of the user, and a model of the user's understanding. The NLG component 212 may generate language for a textual or spoken reply to the user based on these decisions.

[0049] Technical solutions provided by the present inventive subject matter allow users to communicate with an intelligent online personal assistant in a natural conversation. The

assistant is efficient as over time it increasingly understands specific user preferences and is knowledgeable about a wide range of products. Though a variety of convenient input modalities, a user can share photos, or use voice or text, and the assisted user experience may be akin to talking to a trusted, knowledgeable human shopping assistant in a high-end store, for example.

[0050] Conventionally, the approach and data used by online shopping systems aim at a faceless demographic group of buyers with blunt, simplified assumptions to maximize short-term revenue. Conventional sites and apps do not understand how, why, and when users want to be notified. Notifications may be annoying, inappropriate, and impersonal, oblivious to each user's preferences. One person is not the same as a single account. People share accounts and devices. Passwords make platforms neither safe nor easy to use. Problems of weak online identity and the ignoring of environmental signals (such as device, location, notification after anomalous behavior) make it easy to conduct fraud in the marketplace.

[0051] With reference to Figure 3A, the illustrated components of the speech recognition component 210 are now described. A feature extraction component operates to convert raw audio waveform to some-dimensional vector of numbers that represents the sound. This component uses deep learning to project the raw signal into a high-dimensional semantic space. An acoustic model component operates to host a statistical model of speech units, such as phonemes and allophones. These can include Gaussian Mixture Models (GMM) although the use of Deep Neural Networks is possible. A language model component uses statistical models of grammar to define how words are put together in a sentence. Such models can include n-gram-based models or Deep Neural Networks built on top of word embeddings. A speech-to-text (STT) decoder component may convert a speech utterance into a sequence of words typically leveraging features derived from a raw signal using the feature extraction component, the acoustic model component, and the language model component in a Hidden Markov Model (HMM) framework to derive word sequences from feature sequences. In one example, a speech-to-text service in the cloud (e.g., database/cloud 126) has these components deployed in a cloud framework with an API that allows audio samples to be posted for speech utterances and to retrieve the corresponding word sequence. Control parameters are available to customize or influence the speech-to-text process.

[0052] In one example of an artificial intelligence framework 128, two additional parts for the speech recognition component 210 are provided, a speaker adaptation component and a

Language Model (LM) adaptation component. The speaker adaptation component allows clients of an STT system (e.g., speech recognition component 210) to customize the feature extraction component and/or the acoustic model component for each speaker/user. This can be important because most speech-to-text systems are trained on data from a representative set of speakers from a target region and typically the accuracy of the system depends heavily on how well the target speaker matches the speakers in the training pool. The speaker adaptation component allows the speech recognition component 210 (and consequently the artificial intelligence framework 128) to be robust to speaker variations by continuously learning the idiosyncrasies of a user's intonation, pronunciation, accent, and other speech factors, and apply these to the speech-dependent components, e.g,, the feature extraction component, and the acoustic model component. While this approach may require a small voice profile to be created and persisted for each speaker, the potential benefits of accuracy generally far outweigh the storage drawbacks.

[0053] The LM adaptation component operates to customize the language model component and the speech-to-text vocabulary with new words and representative sentences from a target domain, for example, inventory categories or user personas. This capability allows the artificial intelligence framework 128 to be scalable as new categories and personas are supported.

[0054] Figure 3B shows a flow sequence 302 for text normalization in an artificial intelligence framework 128. A text normalization component performing the flow sequence 302 is included in the speech recognition component 210 in one example. Key functionalities in the flow sequence 302 include orthographic normalization (to handle punctuation, numbers, case, and so forth), conversational text normalization (to handle informal chat-type text with acronyms, abbreviations, incomplete fragments, slang, and so forth), and machine translation (to convert a normalized sequence of foreign-language words into a sequence of words in an operating language, including but not limited to English for example).

[0055] The artificial intelligence framework 128 facilitates modern communications. Millennials for example often want to communicate via photos, voice, and text. The technical ability of the artificial intelligence framework 128 to use multiple modalities allows the communication of intent instead of just text. The artificial intelligence framework 128 provides technical solutions and is efficient. It is faster to interact with a smart personal assistant using voice commands or photos than text in many instances.

[0056] Figure 4 is a block diagram illustrating an example software architecture 406, which may be used in conjunction with various hardware architectures described herein. Figure 4 is a non-limiting example of a software architecture and it will be appreciated that many other architectures may be implemented to facilitate the functionality described herein. The software architecture 406 may execute on hardware such as machine 500 of Figure 5 that includes, among other things, processors 504, memory 514, and input/output (I/O) components 518. A representative hardware layer 452 is illustrated and can represent, for example, the machine 500 of Figure 5. The representative hardware layer 452 includes a processing unit 454 having associated executable instructions 404. Executable instructions 404 represent the executable instructions of the software architecture 406, including implementation of the methods, components and so forth described herein. The hardware layer 452 also includes memory and/or storage modules memory/storage 456, which also have executable instructions 404. The hardware layer 452 may also comprise other hardware 458.

[0057] In the example architecture of Figure 4, the software architecture 406 may be conceptualized as a stack of layers where each layer provides particular functionality. For example, the software architecture 406 may include layers such as an operating system 402, libraries 420, applications 416 and a presentation layer 414. Operationally, the applications 416 and/or other components within the layers may invoke application programming interface (API) calls 408 through the software stack and receive a response as in response to the API calls 408. The layers illustrated are representative in nature and not all software architectures have all layers. For example, some mobile or special purpose operating systems may not provide a frameworks/middleware 418, while others may provide such a layer. Other software architectures may include additional or different layers.

[0058] The operating system 402 may manage hardware resources and provide common services. The operating system 402 may include, for example, a kernel 422, services 424 and drivers 426. The kernel 422 may act as an abstraction layer between the hardware and the other software layers. For example, the kernel 422 may be responsible for memory management, processor management (e.g., scheduling), component management, networking, security settings, and so on. The services 424 may provide other common services for the other software layers. The drivers 426 are responsible for controlling or interfacing with the underlying hardware. For instance, the drivers 426 may include display drivers, camera drivers, Bluetooth® drivers, flash memory drivers, serial communication

drivers (e.g., Universal Serial Bus (USB) drivers), Wi-Fi® drivers, audio drivers, power management drivers, and so forth depending on the hardware configuration.

[0059] The libraries 420 provide a common infrastructure that is used by the applications 416 and/or other components and/or layers. The libraries 420 may provide functionality that allows other software components to perform tasks in an easier fashion than to interface directly with the underlying operating system 402 functionality (e.g., kernel 422, services 424, and/or drivers 426). The libraries 420 may include system libraries 444 (e.g., C standard library) that may provide functions such as memory allocation functions, string manipulation functions, mathematical functions, and the like. In addition, the libraries 420 may include API libraries 446 such as media libraries (e.g., libraries to support presentation and manipulation of various known media formats such as MPREG4, H.264, MP3, AAC, AMR, JPG, and PNG), graphics libraries (e.g., an OpenGL framework that may be used to render 2D and 3D graphic content on a display), database libraries (e.g., SQLite that may provide various relational database functions), web libraries (e.g., WebKit that may provide web browsing functionality), and the like. The libraries 420 may also include a wide variety of other libraries 448 to provide many other APIs to the applications 416 and other software components/modules.

[0060] The frameworks frameworks/middleware 418 (also sometimes referred to as middleware) may provide a higher-level common infrastructure that may be used by the applications 416 and/or other software components/modules. For example, the frameworks/middleware 418 may provide various graphic user interface (GUI) functions, high-level resource management, high-level location services, and so forth. The frameworks/middleware 418 may provide a broad spectrum of other APIs that may be utilized by the applications 416 and/or other software components/modules, some of which may be specific to a particular operating system or platform.

[0061] The applications 416 include built-in applications 438 and/or third-party applications 440. Examples of representative built-in applications 438 may include, but are not limited to, a contacts application, a browser application, a book reader application, a location application, a media application, a messaging application, and/or a game application. Third-party applications 440 may include any an application developed using the ANDROID™ or IOS™ software development kit (SDK) by an entity other than the vendor of the particular platform, and may be mobile software running on a mobile operating system such as IOS™, ANDROID™, WINDOWS® Phone, or other mobile operating systems. The third-party applications 440 may invoke the API calls 408 provided

by the mobile operating system (such as operating system 402) to facilitate functionality described herein.

[0062] The applications 416 may use built in operating system functions (e.g., kernel 422, services 424 and/or drivers 426), libraries 420, and frameworks/middleware 418 to create user interfaces to interact with users of the system. Alternatively, or additionally, in some systems interactions with a user may occur through a presentation layer, such as presentation layer 414. In these systems, the application/component "logic" can be separated from the aspects of the application/component that interact with a user.

[0063] Some software architectures use virtual machines. In the example of Figure 4, this is illustrated by a virtual machine 410. The virtual machine 410 creates a software environment where applications/components can execute as if they were executing on a hardware machine (such as the machine 500 of Figure 5, for example). The virtual machine 410 is hosted by a host operating system (operating system (OS) 436 in Figure 4) and typically, although not always, has a virtual machine monitor 460, which manages the operation of the virtual machine as well as the interface with the host operating system (e.g., operating system 402). A software architecture executes within the virtual machine 410 such as an operating system operating system (OS) 436, libraries 434, frameworks 432, applications 430 and/or presentation layer 428. These layers of software architecture executing within the virtual machine 410 can be the same as corresponding layers previously described or may be different.

[0064] Figure 5 is a block diagram illustrating components of a machine 500, according to some example embodiments, which is able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, Figure 5 shows a diagrammatic representation of the machine 500 in the example form of a computer system, within which instructions 510 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 500 to perform any one or more of the methodologies discussed herein may be executed. As such, the instructions may be used to implement modules or components described herein. The instructions transform the general, non-programmed machine into a particular machine programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine 500 operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 500 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or

distributed) network environment. The machine 500 may comprise, but is not limited to, a server computer, a client computer, a personal computer (PC), a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine capable of executing the instructions 510, sequentially or otherwise, that specify actions to be taken by machine 500. Further, while only a single machine 500 is illustrated, the term "machine" will also be taken to include a collection of machines that individually or jointly execute the instructions 510 to perform any one or more of the methodologies discussed herein.

[0065] The machine 500 may include processors 504, memory memory/storage 506, and I/O components 518, which may be configured to communicate with each other such as via a bus 502. The memory/storage 506 may include a memory 514, such as a main memory, or other memory storage, and a storage unit 516, both accessible to the processors 504 such as via the bus 502. The storage unit 516 and memory 514 store the instructions 510 embodying any one or more of the methodologies or functions described herein. The instructions 510 may also reside, completely or partially, within the memory 514, within the storage unit 516, within at least one of the processors 504 (e.g., within the processor's cache memory), or any suitable combination thereof, during execution thereof by the machine 500. Accordingly, the memory 514, the storage unit 516, and the memory of processors 504 are examples of machine-readable media.

[0066] The I/O components 518 may include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. The specific I/O components 518 that are included in a particular machine will depend on the type of machine. For example, portable machines such as mobile phones will likely include a touch input device or other such input mechanisms, while a headless server machine will likely not include such a touch input device. It will be appreciated that the I/O components 518 may include many other components that are not shown in Figure 5. The I/O components 518 are grouped according to functionality merely for simplifying the following discussion and the grouping is in no way limiting. In various example embodiments, the I/O components 518 may include output components 526 and input components 528. The output components 526 may include visual components (e.g., a display such as a plasma display panel (PDP), a light emitting diode (LED) display, a liquid

crystal display (LCD), a projector, or a cathode ray tube (CRT)), acoustic components (e.g., speakers), haptic components (e.g., a vibratory motor, resistance mechanisms), other signal generators, and so forth. The input components 528 may include alphanumeric input components (e.g., a keyboard, a touch screen configured to receive alphanumeric input, a photo-optical keyboard, or other alphanumeric input components), point based input components (e.g., a mouse, a touchpad, a trackball, a joystick, a motion sensor, or other pointing instrument), tactile input components (e.g., a physical button, a touch screen that provides location and/or force of touches or touch gestures, or other tactile input components), audio input components (e.g., a microphone), and the like.

[0067] In further example embodiments, the I/O components 518 may include biometric components 530, motion components 534, environmental environment components 536, or position components 538 among a wide array of other components. For example, the biometric components 530 may include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, or eye tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, or brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, or electroencephalogram based identification), and the like. The motion components 534 may include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environment components 536 may include, for example, illumination sensor components (e.g., photometer), temperature sensor components (e.g., one or more thermometer that detect ambient temperature), humidity sensor components, pressure sensor components (e.g., barometer), acoustic sensor components (e.g., one or more microphones that detect background noise), proximity sensor components (e.g., infrared sensors that detect nearby objects), gas sensors (e.g., gas detection sensors to detection concentrations of hazardous gases for safety or to measure pollutants in the atmosphere), or other components that may provide indications, measurements, or signals corresponding to a surrounding physical environment. The position components 538 may include location sensor components (e.g., a Global Position System (GPS) receiver component), altitude sensor components (e.g., altimeters or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

[0068] Communication may be implemented using a wide variety of technologies. The I/O components 518 may include communication components 540 operable to couple the machine 500 to a network 532 or devices 520 via coupling 522 and coupling 524

respectively. For example, the communication components 540 may include a network interface component or other suitable device to interface with the network 532. In further examples, communication components 540 may include wired communication components, wireless communication components, cellular communication components, Near Field Communication (NFC) components, Bluetooth® components (e.g., Bluetooth® Low Energy), Wi-Fi® components, and other communication components to provide communication via other modalities. The devices 520 may be another machine or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a Universal Serial Bus (USB)).

[0069] Moreover, the communication components 540 may detect identifiers or include components operable to detect identifiers. For example, the communication components processors communication components 540 may include Radio Frequency Identification (RFID) tag reader components, NFC smart tag detection components, optical reader components (e.g., an optical sensor to detect one-dimensional bar codes such as Universal Product Code (UPC) bar code, multi-dimensional bar codes such as Quick Response (QR) code, Aztec code, Data Matrix, Dataglyph, MaxiCode, PDF417, Ultra Code, UCC RSS-2D bar code, and other optical codes), or acoustic detection components (e.g., microphones to identify tagged audio signals). In addition, a variety of information may be derived via the communication components 540, such as, location via Internet Protocol (IP) geo-location, location via Wi-Fi® signal triangulation, location via detecting a NFC beacon signal that may indicate a particular location, and so forth.

[0070] With reference now to Figure 6, an example environment 600 into which an intelligent online personal assistant provided by the intelligent personal assistant system 106 can be deployed is shown. At the center of the environment 600, the intelligent bot 602 with AI appears. The bot leverages the computer vision component 208, the speech recognition component 210, the NLU component 214, the dialog manager 216, the NLG component 212, the search component 220, and identity service 222 to engage users in efficient, interesting, and effective dialog to decode their intent and deliver personalized results.

[0071] An associated application 604 can showcase the bot 602's full power and intelligence with compelling mobile design capabilities and elements. The fabric 606 integrates with Facebook Messenger™, Skype™, and Cortana™ (for example) to enable users to transact where they are already spending time. A smart notifications 610 platform delivers the right information at the right time via any number of channels (e.g., SMS, push notification, email, messaging) to users to encourage them to engage with the bot 602 and

associated marketplaces. Communities 608 features enable users to connect, engage, and interact with their friends, tastemakers, and brands using the same messaging systems in which they already spend most of their time. Other features include group buying and gift buying. A rewards 612 platform incentivizes users to engage more deeply with the bot 602. Rewards can include deep discounts on products, access to unique inventory, and recognition in the app through scores, levels, etc. At marketing 614, a combination of traditional, social and other marketing is performed to win the attention of some populations (e.g., millennials) in more personal ways. Conventional techniques can include merchandising, email, search engine optimization (SEO), and search engine marketing (SEM) as well as experimental techniques such as social ads, viral coupons, and more to target new and existing users.

[0072] FIG. 7 shows an overview of the intelligent personal assistant system 106 processing natural language user inputs to generate an item recommendation in an electronic marketplace. Although the intelligent personal assistant system 106 is not limited to this use scenario, it may be of particular utility in this situation. As previously described, any combination of text, image, and voice data may be received by the artificial intelligence framework 128. Image data may be processed by the computer vision component 208 to provide image attribute data. Voice data may be processed by the speech recognition component 210 into text.

[0073] All of these inputs and others may be provided to the NLU component 214 for analysis. The NLU component 214 may operate to parse user inputs and help determine the user intent and intent-related parameters. For example, the NLU component 214 may discern the dominant object of user interest, and a variety of attributes and attribute values related to that dominant object. The NLU component 214 may also determine other parameters such as the user input type (e.g., a question or a statement) and targeted item recipients. The NLU component 214 may provide extracted data to the dialog manager 216, as well as the AI orchestrator 206 previously shown.

[0074] The NLU component 214 may generally transform formal and informal natural language user inputs into a more formal, machine-readable, structured representation of a user's query. That formalized query may be enhanced further by the dialog manager 216. In one scenario, the NLU component 214 processes a sequence of user inputs including an original query and further data provided by a user in response to machine-generated prompts from the dialog manager 216 in a multi-turn interactive dialog. This user-machine interaction may improve the efficiency and accuracy of one or more automated searches for

the most relevant items available for purchase in an electronic marketplace. The searches may be performed by the search component 220.

[0075] Extracting user intent is very helpful for the AI bot in determining what further action is needed. In one ecommerce-related example, at the very highest level, user intent could be shopping, chit-chat, jokes, weather, etc. If the user intent is shopping, it could relate to the pursuit of a specific shopping mission, gifting an item for a target recipient other than the user, or just to browse an inventory of items available for purchase. Once the high level intent is identified, the artificial intelligence framework 128 is tasked with determining what the user is looking for; that is, is the need broad (e.g., shoes, dresses) or more specific (e.g., two pairs of new black Nike™ size 10 sneakers) or somewhere in between (e.g., black sneakers)?

[0076] In a novel and distinct improvement over the prior art in this field, the artificial intelligence framework 128 may map the user request to certain primary dimensions, such as categories, attributes, and attribute values, that best characterize the available items desired. This gives the bot the ability to engage with the user to further refine the search constraints if necessary. For example, if a user asks the bot for information relating to dresses, the top attributes that need specification might be color, material, and style. Further, over time, machine learning may add deeper semantics and wider "world knowledge" to the system, to better understand the user intent. For example the input "I am looking for a dress for a wedding in June in Italy" means the dress should be appropriate for particular weather conditions at a given time and place, and should be appropriate for a formal occasion. Another example might include a user asking the bot for "gifts for my nephew". The artificial intelligence framework 128 when trained will understand that gifting is a special type of intent, that the target recipient is male based on the meaning of "nephew", and that attributes such as age, occasion, and hobbies/likes of the target recipient should be clarified.

[0077] Figure 8 shows the NLU component 214, its sub-components, and other components with which it interacts, according to some example embodiments. In some embodiments, extracting a user intent is performed by the NLU component 214 by breaking down this often complex technical problem into multiple parts. Each of the various parts of the overall problem of extracting user intent may be processed by particular sub-components of the NLU component 214, sometimes separately and sometimes in combination.

[0078] The sub-components may for example comprise a spelling corrector (speller) 802, a machine translator (MT) 804, a parser 806, a knowledge graph 808, a Named Entity

Recognition (NER) sub-component 810, a Word Sense Detector (WSD) 812, an intent detector 813, and an interpreter 814. The NLU component 214 may receive text, visual selectors, and image attributes, e.g., via the AI orchestrator 206 in one embodiment, and process each separately or in combination. A visual selector is typically a graphical choice provided by a user, such as the selection of a color from a number of presented color samples, or a selection of emoticon that has an associated and thus selected mental state. The NLU component 214 may provide its various outputs, to be described, to the AI orchestrator 206 in one embodiment, to be distributed to other components of the artificial intelligence framework 128 such as the dialog manager 216.

[0079] Other inputs considered by the NLU component 214 may include dialog context 816 (e.g., from context manager 218), user identity information 818 (e.g., from identity service 222), item inventory-related information 820 (e.g., from the core search engine 220 functions of an electronic marketplace), and external world knowledge 822 to improve the semantic inference of user intent from user input. Different types of analyses of these inputs may each yield results that may be interpreted in aggregate and coordinated via the knowledge graph 808. The knowledge graph 808 may for example be based on past users' interactions, inventory-related data, or both.

[0080] The speller 802 may identify and correct spelling mistakes in user-entered text. User text may include, but is not limited to, user queries and item titles. The machine translator 804 may optionally translate user input from the user's natural language into an operating language, including but not limited to English for example. The speller 802 and the machine translator 804 may also coordinate with other normalization sub-components and/or the parser 806 to process abbreviations, acronyms, and slang into more formal data for improved analysis.

[0081] The parser (or dependency parser) 806 may help detect the user's intent by finding a dominant object of the user's input query. This process may involve the parser identifying and analyzing noun-phrases including prepositions and direct and indirect objects, verbs, and affirmations and negations in user input such as from a multi-turn dialog. Affirmations and negations may be detected in the intent detector sub-component 813 in some embodiments, or by different sub-components such as the Word Sense Detector 812.

[0082] In one embodiment, the parser 806 finds the dominant object of user interest from the longest fragment of the user input that can be fully resolved. The parser 806 may also discard user input terms that are of low content, such as "Hi there" and "Can you help me"

and so forth, and/or replace them with less machine-confusing phrases. The parser 806 may also recognize various occasions (e.g., weddings, Mother's Day, and so forth).

[0083] The intent detector 813 may further refine the identification of the user intent by identifying of the dominant objects of interest (which are usually but not always item categories) and the respective best attributes for the results suggested by the parser 806. For example, if the user intent is shopping for a specific item, the knowledge graph 808 may use dominant item categories in a given item inventory (e.g., an eBay inventory, or database/cloud 126) to which it maps. The knowledge graph 808 may also use dominant (e.g., most frequently user-queried or most frequently occurring in an item inventory) attributes pertaining to that item category, and the dominant values for those attributes. Thus, the NLU component 214 may provide as its output the dominant object, user intent, and the knowledge graph 808 that is formulated along dimensions likely to be relevant to the user query. This information may help the dialog manager 216 if there is missing information needed to fully resolve a user query to an item recommendation, and thus whether (and how) to then to prompt the user to further refine the user's requirements via additional input.

[0084] The background information for the knowledge graph 808 may be extracted from the item inventory as a blend of information derived from a hand-curated catalog as well as information extracted from historical user behavior (e.g., a history of all previous user interactions with an electronic marketplace over a period of time). The knowledge graph may also include world knowledge extracted from outside sources, such as internet encyclopedias (e.g., Wikipedia), online dictionaries, thesauruses, and lexical databases (e.g., WordNet). For example, data regarding term similarities and relationships may be available to determine that the terms girl, daughter, sister, woman, aunt, niece, grandmother, and mother all refer to female persons and different specific relative familial relationships. These additional associations may clarify the meaning or meanings of user query terms, and help prevent generation of prompts that may educate the bot but annoy the user. Focus group studies have shown that some users do not want to provide more than a predetermined number, e.g., three, of replies to prompts, so each of those prompts should be as incisive as possible.

[0085] The knowledge graph 808 may be updated dynamically in some embodiments, for example by the AI orchestrator 206. That is, if the item inventory changes or if new user behaviors or new world knowledge data have led to successful user searches, the intelligent online personal assistant 106 is able to take advantage of those changes for future user

searches. An assistant that learns may foster further user interaction, particularly for those users are less inclined toward extensive conversations. Embodiments may therefore modify the knowledge graph 808 may to adjust the information it contains and shares both with other sub-components within the NLU component 214 and externally, e.g. with the dialog manager 216.

[0086] The NER sub-component 810 may extract deeper information from parsed user input (e.g., brand names, size information, colors, and other descriptors) and help transform the user natural language query into a structured query comprising such parsed data elements. The NER sub-component may also tap into world knowledge to help resolve meaning for extracted terms. For example, a query for "a bordeaux" may more successfully determine from an online dictionary and encyclopedia that the query term may refer to an item category (wine), attributes (type, color, origin location), and respective corresponding attribute values (Bordeaux, red, France). Similarly, a place name (e.g., Lake Tahoe) may correspond to a given geographic location, weather data, cultural information, relative costs, and popular activities that may help a user find a relevant item. The structured query depth (e.g., number of tags resolved for a given user utterance length) may help the dialog manager 216 select what further action it should take to improve a ranking in a search performed by the search component 220.

[0087] The Word Sense Detector 812 may process words that are polysemous, that is, have multiple meanings that differ based on the context. For example, the input term "bank" could refer to an "edge of a river" in a geographic sense or a "financial institution" in a purchase transaction payment sense. The Word Sense Detector 812 detects such words and may trigger the dialog manager 216 to seek further resolution from a user if a word sense remains ambiguous. The Word Sense Detector 812 or the intent detector sub-component 813 may also discern affirmations and negations from exemplary phrases including but not limited to "Show me more" or "No, I don't like that", respectively, and so forth. The functions of the parser 804, the intent detector 813, and the Word Sense Detector 812 may therefore overlap or interact to some extent, depending on the particular implementation.

[0088] The interpreter 814 reconciles the analyzed information coming from the various NLU sub-components and prepares output. The output may for example comprise a dominant object of a user query, as well as information resolved regarding relevant knowledge graph dimensions (e.g., item categories, item attributes, item attribute values), the user's intent (e.g., in the case of shopping, whether shopping for a specific item, looking for a gift, or general browsing), a type of user statement recognized, the intended target item

recipient, and so forth. Through the combination of separate analyses performed on shared, augmented, and processed user inputs, the components of the artificial intelligence framework 128 provide a trusted personal shopper (bot) that both understands user intent and is knowledgeable about a wide range of products. The NLU component 214 thus transforms a natural language user query into a structured query to help provide the most relevant results to a user.

[0089] The NLU component 214 therefore improves the operation of the intelligent personal assistant system 106 overall by reducing mistakes, increasing the likelihood of correct divination of user intent underlying a user query, and yielding faster and better targeted searches and item recommendations. The NLU component 214, particularly together with the dialog manager 216 in multi-turn dialog scenarios, effectively governs the operation of the search component 220 by providing more user interaction history-focused and/or item inventory-focused search queries to execute. This distinctive functionality goes beyond the current state of the art via a particular ordered combination of elements as described.

[0090] Examples of use of the NLU component 214 and the intelligent personal assistant system 106 more generally for processing input data from a user are now described. A user may provide a spoken statement like "I am looking for a pair of sunglasses for my wife." The NLU component 214 may process this natural language user input to generate a more formal query to be provided to a search engine 220 and/or dialog manager 216. The more formal query may comprise a group of tags that associate each of one or more resolved handles with a corresponding resolved value. For example, the more formal query may comprise "<intent:gifting, statement-type:statement, dominant-object:sunglasses, target:wife, target-gender:female>". A search engine may provide more relevant results based on a search of these tags than would result from a search of the originally submitted user input.

[0091] In this example, the intelligent personal assistant system 106 determines that the user intent is gifting (versus merely self-shopping or browsing), that the user has provided a statement (versus a question) and that the dominant object of the user's interest is sunglasses. Although the user is shopping, the intent is to gift the item to a particular target item recipient, his wife. A gifting mission is known to be a special type of a shopping mission that may be handled somewhat differently than general inventory browsing or shopping for an item by the user for the user.

[0092] The intelligent personal assistant system 106 may also discern, typically by the Named Entity Recognizer sub-component 810, that "wife" refers to a female person. The particular individual who is the targeted item recipient may be found from data provided by the identity service 212, for example. Further, through use of world knowledge, the intelligent personal assistant system 106 may determine that the term "wife" refers to a married female person, and that children are generally not married. This information may be helpful in constraining a search to women's sunglasses versus other types of sunglasses (e.g., men's sunglasses, children's sunglasses) to generate a more relevant item recommendation without requiring a user prompt to acquire the same information.

[0093] Figure 9 shows the results of various analyses, according to some example embodiments. In one example, a user may type the text input "Hi, can you find me a pair of red nikey shoes?" The resulting formal query may comprise "<intent:shopping, statement-type:question, dominant-object:shoes, target:self, color:red, brand:nike>". Here the user input is a question and the user is shopping for a particular item (versus merely browsing an item inventory or looking for a gift for someone else). The parser 806 may determine that the terms "Hi, can you find me" do not provide a great deal of helpful content and may thus be disregarded.

[0094] The speller sub-component 802 may determine that "nikey" is a known misspelling of the term "nike", and make appropriate correction. The parser sub-component 806 may grammatically analyze the normalized input data by identifying verbs, prepositions, and noun phrases. The grammatical relationships between individual words may illustrate how one word depends on or modifies another, and this information may provide clues for transforming a user query.

[0095] The parser sub-component 806 may also perform noun phrase chunking and discern from the longest parsed query fragment "red nike shoes" that the dominant object of the user's interest is shoes. That is, shoes are determined to be the object of the largest number of modifiers and are at the deepest level of a resulting chunking structure. Note that although the dominant object is often an item category, that is not necessarily always the case. The dominant object here is also described by modifiers ("red" and "nike") which the Named Entity Recognizer 810 may determine relate to a color and a brand, respectively.

[0096] Note also that in this instance two attributes (color, brand) and corresponding attribute values (red, nike) are provided for the dominant object, while in the previous example at best one attribute was provided (e.g., women's sunglasses were indirectly

specified via inference). The dialog manager 216 may decide as a result that the user's original query is sufficiently constrained that an appropriate prompt may be one or more item recommendations, rather than a question asking a user for additional constraints that would further narrow the subsequent search. In contrast, for the previous query much more detail regarding women's sunglasses may be needed, so the dialog manager 216 may generate a number of prompts in a multi-turn dialog to that end. Some users are annoyed by a large number of prompts however, and would prefer to deal with a bot that is able to extract more information on its own from every turn. It is therefore advantageous to minimize the number of turns in a multi-turn dialog by gleaning as much information from each user utterance as possible.

[0097] For example, the NLU component 214 may determine that there are many different listings for red nike shoes in a searched item inventory and/or that the interactions of previous users determined additional attribute values before users made item selections. Therefore, the NLU component 214 may consult the knowledge graph 808 to determine the most helpful attributes for this dominant object of user interest. The knowledge graph 808 may have information indicating that for the item category "shoes", the most helpful and/or frequently specified attributes are color, brand, and size, along with corresponding conditional probability values showing the relative correlation or association strength or conditional probability of importance of each in finding a relevant item. It may be the case that all of those attributes probably need to be parameterized for a query to be deemed sufficiently specific to result in search success. It may also be the case however that only a limited number of the attributes that adequately cover a predetermined percentage of the available associations need to be parameterized.

[0098] The user has provided attribute values for color and brand in this example, but not for size, so the dialog manager 216 may therefore ask the user "What size do you want?" and await further user input. Suppose the user replies "I want 10." What does this mean? The intelligent personal assistant system 106 could interpret "I want 10" as meaning the user wants ten of the previously specified red nike shoes. World knowledge might provide the information that shoes generally come in pairs, so a reinterpretation of the user's response to the prompt could be somewhat refined to the notion that the user instead wants ten pairs of red nike shoes. Neither interpretation is correct however, because neither considers the context of the conversation. That is, the "I want 10" user input is a reply to a prompt that was generated to gather more information (a value for the size attribute in this case) regarding a previous utterance. If the intelligent personal online assistant 106 cannot

associate the reply with any previous user inputs, it may output an error statement indicating that it cannot resolve the conversational context.

[0099] The context manager 218 may prevent such confusion by tracking not only the long-term history of user interactions but also the short-term memory of a current user's interactions for a given shopping mission. A reply to a prompt in a multi-turn dialog is not necessarily an isolated user utterance, but is usually contextually related to previous user utterances and previous prompts (if any) in a dialog. The intelligent personal assistant system 106 is therefore adapted toward user conversations that lead to accumulated search constraints sufficient to make a refined search query more successful at finding a relevant item to recommend.

[00100] In some cases however, the NLU component 214 may determine that the user has abandoned a previous query mission and is now interested in finding something else. The dialog manager 216 may therefore receive an indication of that determination from the NLU component 214 in some embodiments, and alter its behavior accordingly. That dialog manager 216 behavior may comprise saving the interactions for the current search mission for possible later use for example, and starting a new dialog based on the current user utterance without using any of the context information relating to the previous search mission. In one embodiment, the NLU component 214 may determine that such a change of mission has occurred when a new dominant object of user interest is detected.

[00101] Figure 10 shows a knowledge graph 808, according to some example embodiments. The knowledge graph 808 is generally a database or file that represents a plurality of nodes, shown here in ellipses. Each node may represent an item category, an item attribute, or an item attribute value for the exemplary scenario of processing natural language user inputs to generate an item recommendation. In this example, item categories include "Men's Athletic Shoes", "Cars & Trucks", and "Women's Athletic Shoes". Each item category may have been assigned an identification number, as shown, by an inventory tracking system or by the intelligent personal assistant system 106 for example.

[00102] The item attributes shown in the knowledge graph 808 in this example include "Product Line", "Brand", "Color", and "Style". Item attributes are often directly linked to item categories, although that is not always the case. The item attribute values shown in the knowledge graph 808 in this example include "Air Jordan", "Kobe Bryant", "Air Force 1", "Asics", "Nike", "New Balance", "Adidas", "Blue", "White", "Red", "Black", "Metallic

Black", "Running", "Basketball", and "Sneakers". The item attribute values are often directly linked to item attributes, although again that is not always the case.

[00103] The links shown between the knowledge graph 808 nodes are directed edges that may have an associated correlation or association value indicating a strength of a relationship between two particular nodes. Some of the correlation values of the knowledge graph 808 are indicated in Figure 10. The correlation values may be created in a variety of ways, and may be used for a variety of purposes.

[00104] For example, in one embodiment, the correlation values may be derived from an inventory of items available for purchase. The inventory may be current or historical. When a seller lists an item for sale, the seller may specify one or more item categories, attributes, and/or attribute values as metadata that describe the item and are thus useful search terms that may be provided by a user who is looking to buy the item. In some cases, an electronic marketplace may categorize a seller's item in various ways, such as by providing guides to a seller that describe available predetermined item categories and commonly used descriptive terms.

[00105] For example, a seller may have a pair of shoes to sell and may specify that they are men's blue athletic running shoes made by Adidas. The seller may specify to the marketplace that the item category is "men's athletic shoes", and the seller may be prompted to specify item attributes, for example from a list of item attributes. Alternately, an electronic marketplace may identify that the seller has provided a number of item attribute values, and may automatically relate these item attribute values to various item attributes, e.g., those attributes that have those values as specified possibilities, perhaps in metadata. The electronic marketplace may for example determine that "men's athletic shoes" is actually a sub-category or attribute of the broader category of "shoes" because either a seller or the electronic marketplace for example has defined sub-categories or attributes for that category.

[00106] An electronic marketplace may periodically analyze its inventory of items available for sale and provide summary data describing that inventory in the form of the knowledge graph 808. In this approach, the exemplary knowledge graph 808 may note that of all inventory items in the category "men's athletic shoes", thirty percent (or 0.3) of the items are characterized by the item attribute "Product Line", forty percent (or 0.4) of the items are characterized by the item attribute "Brand", and twenty percent (0.2) of the items are characterized by the item attribute "Color", as shown. Of the items characterized by the

item attribute "Product Line", twenty percent (or 0.2) have the item attribute value of "Kobe Bryant" and ten percent (or 0.1) have the item attribute value of "Air Force 1", as shown. Thus, in this embodiment, the knowledge graph 808 may comprise entries that describe the actual inventory of items available.

[00107] For a very large electronic marketplace with perhaps millions of items available for purchase, detailed analysis of the entire item inventory, particularly its status at any given moment in time, may be computationally expensive. Such analyses may therefore be performed only on an occasional or periodic ongoing basis. Statistical sampling methods may also produce a knowledge graph 808 that describes an approximate estimate of the characteristics of the item inventory.

[00108] During the processing of a user query, the parsed input data elements from the user query may be matched against the dimensions of the knowledge graph 808 to help match the user's demands with the available supply of items. The dimensions of the knowledge graph 808 may comprise the item categories, item attributes, and item attribute values that describe the items available for purchase. If a user has expressed an interest in men's athletic shoes, the user expects the intelligent personal assistant system 106 to help the user find a relevant item from the inventory of items available for purchase. Finding an item that is not available for purchase may cause a user to lose interest in shopping entirely, an outcome of great concern. The correlational values may therefore represent the relative number of items in a given item category, described by a given item attribute, or having a given item attribute value. The correlational values may be based on conditional probabilities, e.g. given that a particular item attribute is specified, what is the probability that a particular item attribute value is specified.

[00109] In a different embodiment, the knowledge graph 808 may be based on the historical interaction of all users with an electronic marketplace over a period of time. That is, the nodes may comprise search terms provided by many previous users in their utterances or navigational histories with the marketplace. Rather than analyzing the inventory as previously described, this approach analyzes the user behavior, e.g. what buyers are saying and doing when they are interacting with the marketplace to find a relevant item in the inventory.

[00110] In this example embodiment, the correlation values shown in Figure 10 may indicate the most prevalent or frequently occurring user interactions in terms of conditional probabilities. If a user indicates an interest in women's athletic shoes for example, the

knowledge graph 808 may indicate that in thirty percent (or 0.3) of such buyer interactions, the buyer specifies an item attribute of "Style", in twenty percent (or 0.2) of such buyer interactions, the buyer specifies an item attribute of "Brand", and in thirty percent (0.3) of such buyer interactions, the buyer specifies an item attribute of "Color". Thus, regardless of the available inventory, the knowledge guide 808 characterizes the search behavior of users, e.g., how users are attempting to find relevant items.

[00111] As in the previous embodiment, during the processing of a user query, the parsed input data elements from the user query may be matched against the dimensions of the knowledge graph 808 to help match the user's demands with the available supply of items. However, the dimensions of the knowledge graph 808 may now comprise the categories, attributes, and attribute values provided by previous users' query inputs when they were looking for relevant items to purchase. If a user has expressed an interest in women's athletic shoes for example, the intelligent personal assistant system 106 may best proceed by determining how other users went about finding an item relevant to women's athletic shoes item from the inventory of items available for purchase. The correlation values or scores in the knowledge graph 808 may therefore represent the relative number of times a given category, a given attribute, or a given attribute value were used in leading to a satisfactory search conclusion. The correlation values in other words may effectively represent a measure of how "beaten" is a given portion of a user interaction path traversing from one search term to another in the knowledge graph 808.

[00112] Regardless of how the knowledge graph 808 is formulated, the match between user input terms and knowledge graph dimensions (e.g., item categories, item attributes, and item attribute values) may be used to transform the original user query into an improved query. The match may for example help determine what, if any, prompts should be generated for the user in a multi-turn dialog to best find a relevant search result. Thus, the NLU component 214 may provide information from the knowledge graph 808 to the dialog manager 216 for this purpose. That is, the NLU component 214 may deliver a concise knowledge graph 808, with dimensions having some relevance, to the dialog manager 216, along with the dominant object of user interest, user intent, and related parameters.

[00113] Figures 11A and 11B show a concise knowledge graph 808 with an item category, some item attributes, and some item attribute values, according to some example embodiments. Each figure is shown and discussed separately for clarity, but together may refer actually to a knowledge graph 808 common to the two figures. In Figure 11A, the normalized and parsed user query has provided the item attribute/value tags of <color:red,

brand:nike> for a dominant object of user interest "Shoes", as previously described. The knowledge graph 808 indicates there is a forty percent (0.4) correlation between "Shoes" and "Men's Athletic Shoes", and that there is a forty percent (0.4) correlation between "Men's Athletic Shoes" and "Brand", and a twenty percent (0.2) correlation between "Men's Athletic Shoes" and "Color". There is also a thirty percent (0.3) correlation between "Men's Athletic Shoes" and "Product Line", and various correlations for various item attribute values (e.g., "Air Jordan", "Kobe Bryant", and "Air Force 1") are known. Thus, whether based on inventory or user behavior, the as-yet unspecified query terms of "Men's Athletic Shoes" and "Product Line" have significant associations with a successful search. The dialog manager 216 may therefore rank and prioritize the parameterization of these as-yet unspecified possibilities through user prompts according to their association or correlation values, or their relative positions in the knowledge graph 808 hierarchy, or a combination of both.

[00114] Similarly, for Figure 11B, the knowledge graph 808 indicates there is a thirty percent (0.3) correlation between "Shoes" and "Women's Athletic Shoes", and that there is a thirty percent (0.4) correlation between "Women's Athletic Shoes" and "Style". Neither "Women's Athletic Shoes" nor "Style" have been specified by the user, nor have relevant item attribute values for "Style" (e.g., "Basketball", "Running", and "Sneakers") been specified. The dialog manager 216 may therefore also prioritize the parameterization of these as-yet unspecified possibilities through user prompts according to their association or correlation values, or their relative positions in the knowledge graph 808 hierarchy, or a combination of both.

[00115] In one prompt generation strategy, the dialog manager 216 may proceed from the broadest category to a sub-category or attribute and then to an attribute value to determine a sequence of prompt topics, in that order. That is, given that the category of "Shoes" has been specified, the dialog manager 216 may proceed directly to resolve whether the user is interested in "Men's Athletic Shoes" or "Women's Athletic Shoes" as those two possibilities have the highest (or only) available association strengths in the knowledge graph 808. This hierarchically guided search approach may appeal to users who do not want to answer more than a limited number of prompts to zero in on a relevant item.

[00116] In another prompt generation strategy, the dialog manager 216 may choose prompt topics more randomly from all unspecified attributes and attribute values that appear in the knowledge graph 808. Although this approach is somewhat undirected, it may be appropriate when a user is browsing an inventory, versus pursuing a specific shopping

mission. Users who are not annoyed by chatting with an intelligent personal assistant system 106 may prefer this more exploratory or conversational approach that in a sense wanders through the possibilities of the knowledge graph 808.

[00117] In Figures 11A and 11B, candidate prompts for further user input may be selected by whether the user is more interested in men's or women's athletic shoes, and also, accordingly, whether the user is interested in a particular product line or style. Note that the narrower attributes in the knowledge graph 808 (product line or style, in this case) may actually be better candidates for a user prompt in some situations, depending on how dispositive each candidate is. That is, style and product line are each equally associated with the respective item attribute or sub-category above each of them in the knowledge graph 808, but there is more data available for the product line attribute value possibilities. Thus, a prompt that asks if a user is interested in Air Jordan shoes implicitly also asks if the user is interested in a particular product line and in men's athletic shoes. A single affirmation or negation by the user could therefore help discern the user's intent in terms accepting or rejecting multiple possibilities (e.g. attribute and attribute value) at one time.

[00118] Figure 12 shows an overview of the intelligent personal assistant system 106 processing natural language user inputs to generate suggestive prompts, according to some example embodiments. Since prompts to users that are not incisive (e.g., providing information that could be determined without asking the user) are known to annoy some users, some embodiments may use additional data to narrow the field of possible search constraints to be expressly given by a user. For example, the NLU component 214 has discerned the user is interested in shopping for red nike shoes, and the knowledge graph 808 indicates that men's athletic shoes and women's athletic shoes are possible prompt subjects (among others).

[00119] However, additional data may be available that indicates whether the user is interested in men's athletic shoes or women's athletic shoes without asking. For example, the current user's interaction history with the electronic marketplace may indicate that most or all of the user's purchases have been for items associated with women. That may be because the current user is a woman performing another self-shopping mission, or perhaps because the current user often performs gifting missions where the intended target recipient is a woman, for example. Further, world knowledge or other potentially relevant external contextual information may adjust the weighting of prompt possibilities by dialog manager 216. External data regarding locations, weather, costs, culture, and occasions for example

may play similar roles in adjusting the determination of a next prompt for maximum incisiveness.

[00120] The intelligent personal assistant system 106 may therefore conclude that the user is probably more interested in women's athletic shoes than men's athletic shoes without generating a prompt to confirm that point. The dialog manager 216 may thus proceed to the next most likely-incisive prompt topic based on the processed user inputs and the knowledge graph 808. In the example of Figure 11B, given that the user is interested in women's athletic shoes and has already specified values for the attributes of brand and color, the best candidate prompt may relate to the as-yet unspecified attribute, style.

[00121] The dialog manager 216 may therefore simply ask the user "What type of style do you prefer?" However, this approach does not take advantage of the additional knowledge available regarding item attribute values in the knowledge graph 808, whether from item inventory data or past user interaction data. Therefore, in one embodiment, the dialog manager may generate a prompt for additional user input that also states alternatives that are available in the knowledge graph 808 and/or may have association values available in the knowledge graph 808.

[00122] For example, prompt 1202 may instead ask the user "What type of style do you prefer, such as sneakers or running shoes?" This type of question prompt formulation both informs the user of suggestions that may be relevant (e.g. due to inventory or past user interaction behaviors) to a successful search and gathers additional user input. Note that not all of the known item attribute values in the knowledge graph need be suggested, and not all edges directed between entries may have a specified score value. As before, the intelligent personal assistant system 106 may use other data to winnow the possibilities to those that are more discerning.

[00123] Further, the dialog manager 216 may even provide suggested precise user input phrasing that is likely to lead to a relevant search result when used in a reply. For example, prompt 1202 may instead ask the user "Would you like 'sneaker style' or 'running shoe style'?". Such phrasing suggestions may lead to reply (particularly a spoken reply) that has all of the remaining as-yet-unspecified constraints in an easily processed form (e.g., "sneaker style" specifies both an attribute value of sneaker and an attribute of style).

[00124] In another example, the dialog manager 216 may have enough data from the analysis of the user inputs and from other data to generate a prompt that makes suggestive item recommendations. In this case, the dialog manager may have data indicating that the

user may be interested in sneakers. Rather than using a question type prompt to directly confirm that, the dialog manager 216 may proceed with a search and output text and/or images of a few possibly relevant inventory items to the user. Prompt 1204 may thus announce "I found these sneakers:" and show images of (or, more generally, images characterizing) specific items or item groups available for purchase. This approach makes it easy for a user who provided a less than fully-constrained query to affirm or negate a single suggestion type prompt. The affirmation may be verbal reply or a selection of a particular displayed item, for example.

[00125] In another example, the dialog manager 216 may select a prompt that comprises a validating statement, such as "I understand you want to find red nike shoes" or "OK I can help you find red nike shoes now" to conversationally lead the user to provide further confirmatory and revelatory discussion of the dominant object of user interest. This prompt type allows a user to resolve ambiguities that the intelligent personal assistant system 106 may not have been able to resolve automatically without asking question type prompts that may cause confusion. This ambiguity may occur for example if there are many unusual spelling errors in user textual input, or if the user's speech was received in a noisy environment, so that normalizing has not worked well.

[00126] The validating statement type prompt may also be of particular utility when a user has provided an utterance that indicates a change in user interest. That is, the bot may make a validating statement to allow the user to confirm that a new search mission has begun, and that the context of a previous search mission is no longer applicable. For example, the bot that was previously looking for red nike shoes may respond to a user input regarding an umbrella with "OK let's look for an umbrella now instead of red nike shoes." If the user did not intend to change interest, there is a good chance the user will provide a more detailed reply summarizing the relevant query terms to put the bot "back on target".

[00127] In another example, the dialog manager 216 may generate a prompt that not only indicates that no item meeting all of the specified search constraints has been found in an inventory, but that items meeting some or most of the specified search constraints have been found via a search to be available. For example, if no red nike shoes of the user query are in the inventory, the dialog manager 216 may say "No red nike shoes are currently available, but nike shoes are available now in blue or green." This prompt approach thus avoids the dead-end search outcome that might cause a user to lose interest in searching entirely, and encourages the user to pursue a slightly broadened or modified search that is already determined to be likely to succeed. The dialog manager 216 may thus encourage a user to

"backtrack" and continue searching via related item attribute values, item attributes, or even item categories. This prompt generation approach may be of particular utility for someone who is browsing or searching for a gift for a target recipient whose preferences are not well known.

[00128] Similarly, if a user is looking for black nike shoes but only red, blue, and green nike shoes are available in the inventory as determined by a search, a prompt that asks the user if the user is interested in black nike shoes may be counterproductive and actually annoying. Therefore, in one embodiment, no prompt of any type is generated by the dialog manager 216 if such a prompt, when affirmed by the user's reply, will lead to items that are not available an inventory. That is, this version of the intelligent online personal assistant 106 does not actively lead the user into a dead end.

[00129] Figure 13 shows a flowchart of a methodology for processing natural language user inputs to generate an item recommendation, according to some example embodiments. This methodology may be implemented via the structural elements previously described, as well as via instructions executed by a processor in a computing machine. At 1302, the methodology may receive input data from a user. At 1304, the methodology may normalize the received input data. At 1306, the methodology may parse the normalized input data, to for example identify a dominant object of user interest and related parameters from the parsed input data.

[00130] At 1308, the methodology may analyze the parsed input data to find matches between the dimensions of the knowledge graph 808 and the dominant object and the related parameters. At 1310, the methodology may aggregate the analysis results into a formal query for searching. At 1312, the methodology may optionally generate a user prompt or prompts for additional input data from the user.

[00131] Although the subject matter has been described with reference to specific example embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader scope of the disclosed subject matter. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense. The accompanying drawings that form a part hereof, show by way of illustration, and not of limitation, specific embodiments in which the subject matter may be practiced. The embodiments illustrated are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed herein. Other embodiments may be utilized and derived therefrom, such that structural and logical substitutions and changes

may be made without departing from the scope of this disclosure. This Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by any appended claims, along with the full range of equivalents to which such claims are entitled.

[00132] Such embodiments of the inventive subject matter may be referred to herein, individually and/or collectively, by the term "invention" merely for convenience and without intending to voluntarily limit the scope of this application to any single invention or inventive concept if more than one is in fact disclosed. Thus, although specific embodiments have been illustrated and described herein, it should be appreciated that any arrangement calculated to achieve the same purpose may be substituted for the specific embodiments shown. This disclosure is intended to cover any and all adaptations or variations of various embodiments. Combinations of the above embodiments, and other embodiments not specifically described herein, will be apparent to those of skill in the art upon reviewing the above description.

[00133] The following numbered examples are embodiments.

[00134] 1. A method for generating a prompt for additional natural language input in a multi-turn dialog, the method comprising:

receiving ranked matches between dimensions in a knowledge graph and the results of an analysis of user query data, the knowledge graph dimensions comprising at least one each of a category, an attribute, and an attribute value, and the results comprising a dominant object of user interest, user intent, and related parameters;

searching an inventory and incorporating search results into the knowledge graph;

determining if a predetermined sufficient level of matching between the results of the analysis and knowledge graph dimensions linked, directly or indirectly, to the dominant object has been achieved; and

if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt based on at least one unspecified linked knowledge graph dimension.

[00135] 2. The method of example 1, further comprising suppressing the question prompt if no reply could lead to a satisfactory search result.

[00136] 3. The method of example 1 or example 2, wherein the question prompt suggests a choice of linked knowledge graph dimensions based on association strength values.

[00137] 4. The method of any one of examples 1 to 3, wherein the question prompt provides suggested reply phrasing based on at least one of the linked knowledge graph dimensions.

[00138] 5. The method of any one of examples 1 to 4, further comprising instead generating a different question prompt that notes knowledge graph dimensions that do not meet all user search constraints if there are no knowledge graph dimensions that meet all user search constraints.

[00139] 6. The method of any one of examples 1 to 5, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of knowledge graph element association strength values, and a predetermined order of knowledge graph dimension types.

[00140] 7. The method of any one of examples 1 to 6, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of: a depth of linked knowledge graph data elements that can be resolved by an affirmation or negation type reply, and a degree of randomness in unspecified knowledge graph dimension selection.

[00141] 8. A computer-readable storage medium having embedded therein a set of instructions which, when executed by one or more processors of a computer, causes the computer to execute the following operations for generating a prompt for additional natural language input in a multi-turn dialog, the operations comprising:

receiving ranked matches between dimensions in a knowledge graph and the results of an analysis of user query data, the knowledge graph dimensions comprising at least one each of a category, an attribute, and an attribute value, and the results comprising a dominant object of user interest, user intent, and related parameters;

searching an inventory and incorporating search results into the knowledge graph;

determining if a predetermined sufficient level of matching between the results of the analysis and knowledge graph dimensions linked, directly or indirectly, to the dominant object has been achieved; and

if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt based on at least one unspecified linked knowledge graph dimension.

[00142] 9. The medium of example 8, further comprising suppressing the question prompt if no reply could lead to a satisfactory search result.

[00143] 10. The medium of example 8 or example 9, wherein the question prompt suggests a choice of linked knowledge graph dimensions based on association strength values.

[00144] 11. The medium of any one of examples 8 to 10, wherein the question prompt provides suggested reply phrasing based on at least one of the linked knowledge graph dimensions.

[00145] 12. The medium of any one of examples 8 to 11, further comprising instead generating a different question prompt that notes knowledge graph dimensions that do not meet all user search constraints if there are no knowledge graph dimensions that meet all user search constraints.

[00146] 13. The medium of any one of examples 8 to 12, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of knowledge graph element association strength values, and a predetermined order of knowledge graph dimension types.

[00147] 14. The medium of any one of examples 8 to 13, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of: a depth of linked knowledge graph data elements that can be resolved by an affirmation or negation type reply, and a degree of randomness in unspecified knowledge graph dimension selection.

[00148] 15. A system to generate a prompt for additional natural language input in a multi-turn dialog, the system comprising:

a natural language understanding component configured to provide ranked matches between dimensions in a knowledge graph and the results of an analysis of user query data, the knowledge graph dimensions comprising at least one each of a category, an attribute, and an attribute value, and the results comprising a dominant object of user interest, user intent, and related parameters;

a search component configured to search an inventory and incorporating search results into the knowledge graph;

a dialog manager component configured to determine if a predetermined sufficient level of matching between the results of the analysis and knowledge graph dimensions linked, directly or indirectly, to the dominant object has been achieved; and

if the sufficient level of matching has not been achieved, then generating and outputting with the dialog manager component a question type prompt based on at least one unspecified linked knowledge graph dimension.

[00149] 16. The system of example 15, wherein the question prompt is suppressed if no reply could lead to a satisfactory search result.

[00150] 17. The system of example 15 or example 16, wherein the question prompt suggests a choice of linked knowledge graph dimensions based on association strength values.

[00151] 18. The system of any one of examples 15 to 17, wherein the question prompt provides suggested reply phrasing based on at least one of the linked knowledge graph dimensions.

[00152] 19. The system of any one of examples 15 to 18, further comprising instead generating with the dialog manager component a different question prompt that notes knowledge graph dimensions that do not meet all user search constraints if there are no knowledge graph dimensions that meet all user search constraints.

[00153] 20. The system of any one of examples 15 to 19, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting with the dialog manager component a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of knowledge graph element association strength values, a predetermined order of knowledge graph dimension types, a depth of linked knowledge graph data elements that can be resolved by an affirmation or negation type reply, and a degree of randomness in unspecified knowledge graph dimension selection.

[00154] 21. A machine-readable medium carrying a set of instructions which, when executed by one or more processors of a computer, causes the computer to carry out the method of any one of examples 1 to 7.

CLAIMS

1. A method for generating a prompt for additional natural language input in a multi-turn dialog, the method comprising:

receiving ranked matches between dimensions in a knowledge graph and the results of an analysis of user query data, the knowledge graph dimensions comprising at least one each of a category, an attribute, and an attribute value, and the results comprising a dominant object of user interest, user intent, and related parameters;

searching an inventory and incorporating search results into the knowledge graph;

determining if a predetermined sufficient level of matching between the results of the analysis and knowledge graph dimensions linked, directly or indirectly, to the dominant object has been achieved; and

if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt based on at least one unspecified linked knowledge graph dimension.

2. The method of claim 1, further comprising suppressing the question prompt if no reply could lead to a satisfactory search result.

3. The method of claim 1, wherein the question prompt suggests a choice of linked knowledge graph dimensions based on association strength values.

4. The method of claim 1, wherein the question prompt provides suggested reply phrasing based on at least one of the linked knowledge graph dimensions.

5. The method of claim 1, further comprising instead generating a different question prompt that notes knowledge graph dimensions that do not meet all user search constraints if there are no knowledge graph dimensions that meet all user search constraints.

6. The method of claim 1, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of knowledge graph element association strength values, and a predetermined order of knowledge graph dimension types.

7. The method of claim 1, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of: a depth of linked knowledge graph data elements that can be resolved by an affirmation or negation type reply, and a degree of randomness in unspecified knowledge graph dimension selection.

8. A computer-readable storage medium having embedded therein a set of instructions which, when executed by one or more processors of a computer, causes the computer to execute the following operations for generating a prompt for additional natural language input in a multi-turn dialog, the operations comprising:

receiving ranked matches between dimensions in a knowledge graph and the results of an analysis of user query data, the knowledge graph dimensions comprising at least one each of a category, an attribute, and an attribute value, and the results comprising a dominant object of user interest, user intent, and related parameters;

searching an inventory and incorporating search results into the knowledge graph;

determining if a predetermined sufficient level of matching between the results of the analysis and knowledge graph dimensions linked, directly or indirectly, to the dominant object has been achieved; and

if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt based on at least one unspecified linked knowledge graph dimension.

9. The medium of claim 8, further comprising suppressing the question prompt if no reply could lead to a satisfactory search result.

10. The medium of claim 8, wherein the question prompt suggests a choice of linked knowledge graph dimensions based on association strength values.

11. The medium of claim 8, wherein the question prompt provides suggested reply phrasing based on at least one of the linked knowledge graph dimensions.

12. The medium of claim 8, further comprising instead generating a different question prompt that notes knowledge graph dimensions that do not meet all user search constraints if there are no knowledge graph dimensions that meet all user search constraints.

13. The medium of claim 8, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of knowledge graph element association strength values, and a predetermined order of knowledge graph dimension types.

14. The medium of claim 8, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of: a depth of linked knowledge graph data elements that can be resolved by an affirmation or negation type reply, and a degree of randomness in unspecified knowledge graph dimension selection.

15. A system to generate a prompt for additional natural language input in a multi-turn dialog, the system comprising:

a natural language understanding component configured to provide ranked matches between dimensions in a knowledge graph and the results of an analysis of user query data, the knowledge graph dimensions comprising at least one each of a category, an attribute, and an attribute value, and the results comprising a dominant object of user interest, user intent, and related parameters;

a search component configured to search an inventory and incorporating search results into the knowledge graph;

a dialog manager component configured to determine if a predetermined sufficient level of matching between the results of the analysis and knowledge graph dimensions linked, directly or indirectly, to the dominant object has been achieved; and

if the sufficient level of matching has not been achieved, then generating and outputting with the dialog manager component a question type prompt based on at least one unspecified linked knowledge graph dimension.

16. The system of claim 15, wherein the question prompt is suppressed if no reply could lead to a satisfactory search result.

17. The system of claim 15, wherein the question prompt suggests a choice of linked knowledge graph dimensions based on association strength values.

18. The system of claim 15, wherein the question prompt provides suggested reply phrasing based on at least one of the linked knowledge graph dimensions.

19. The system of claim 15, further comprising instead generating with the dialog manager component a different question prompt that notes knowledge graph dimensions that do not meet all user search constraints if there are no knowledge graph dimensions that meet all user search constraints.

20. The system of claim 15, further comprising that if the sufficient level of matching has not been achieved, then generating and outputting with the dialog manager component a question type prompt regarding linked unspecified knowledge graph dimensions based on at least one of knowledge graph element association strength values, a predetermined order of knowledge graph dimension types, a depth of linked knowledge graph data elements that can be resolved by an affirmation or negation type reply, and a degree of randomness in unspecified knowledge graph dimension selection.

21. A machine-readable medium carrying a set of instructions which, when executed by one or more processors of a computer, causes the computer to carry out the method of any one of claims 1 to 7.

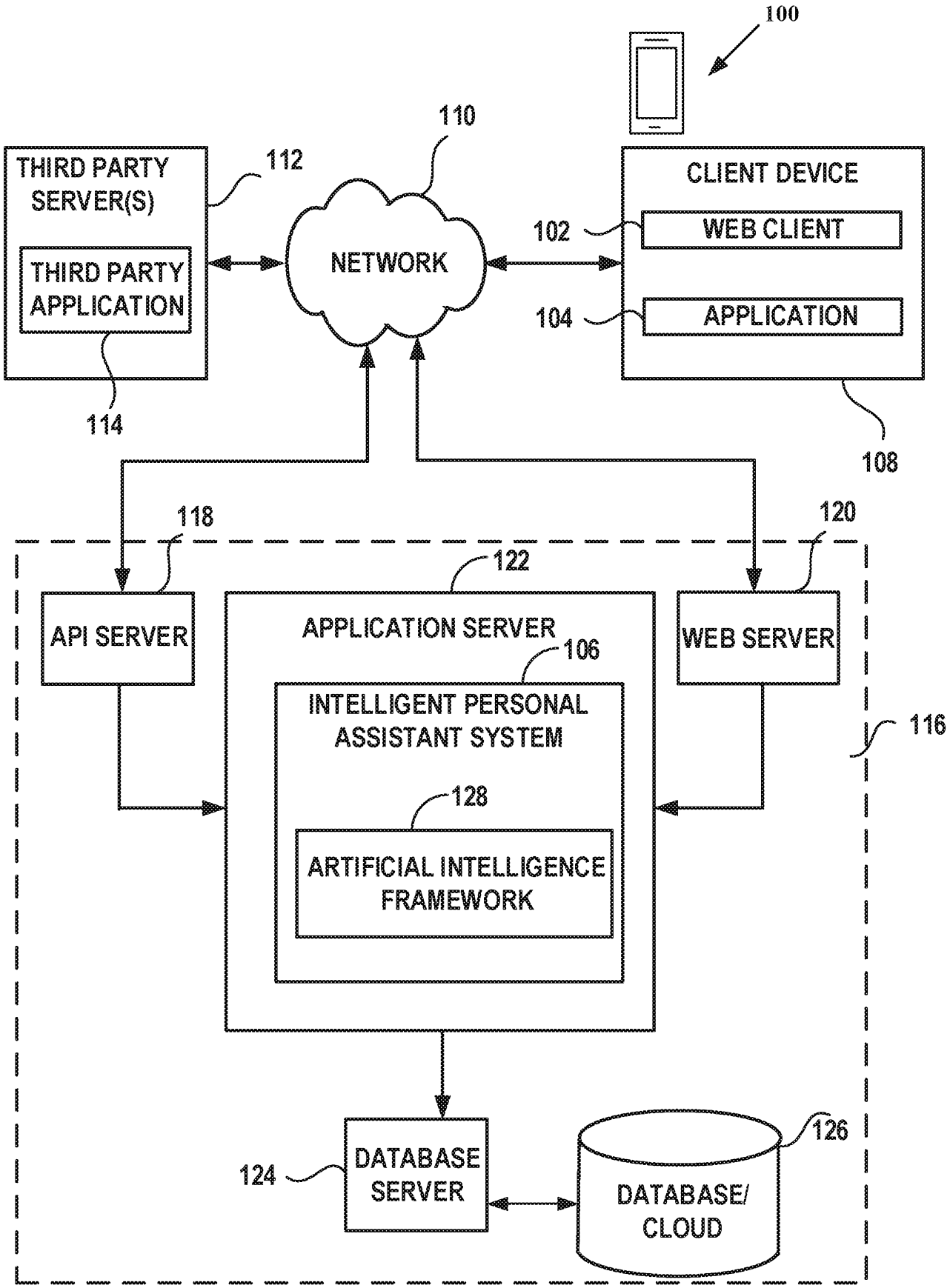
100
110
THIRD PARTY SERVER(S)
112
THIRD PARTY APPLICATION
114
NETWORK
CLIENT DEVICE
102
WEB CLIENT
104
APPLICATION
108
118
120
122
API SERVER
APPLICATION SERVER
106
WEB SERVER
INTELLIGENT PERSONAL ASSISTANT SYSTEM
116
128
ARTIFICIAL INTELLIGENCE FRAMEWORK
124
DATABASE SERVER
126
DATABASE/ CLOUD

FIG. 1

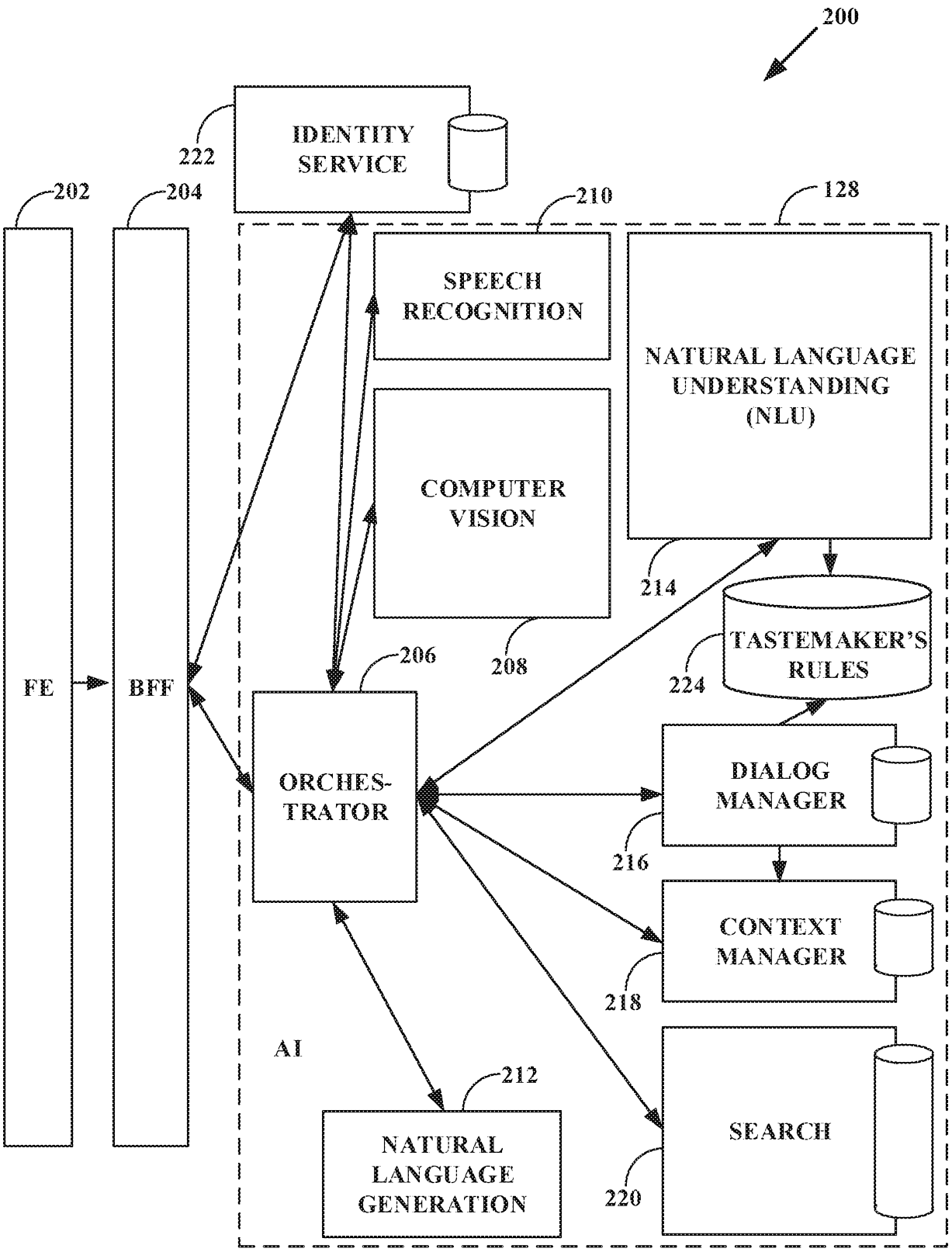
200
IDENTITY SERVICE
222
202
204
210
128
SPEECH RECOGNITION
NATURAL LANGUAGE UNDERSTANDING (NLU)
COMPUTER VISION
214
TASTEMAKER'S RULES
224
208
206
FE
BFF
ORCHES-TRATOR
DIALOG MANAGER
216
CONTEXT MANAGER
218
AI
212
NATURAL LANGUAGE GENERATION
SEARCH
220

FIG. 2

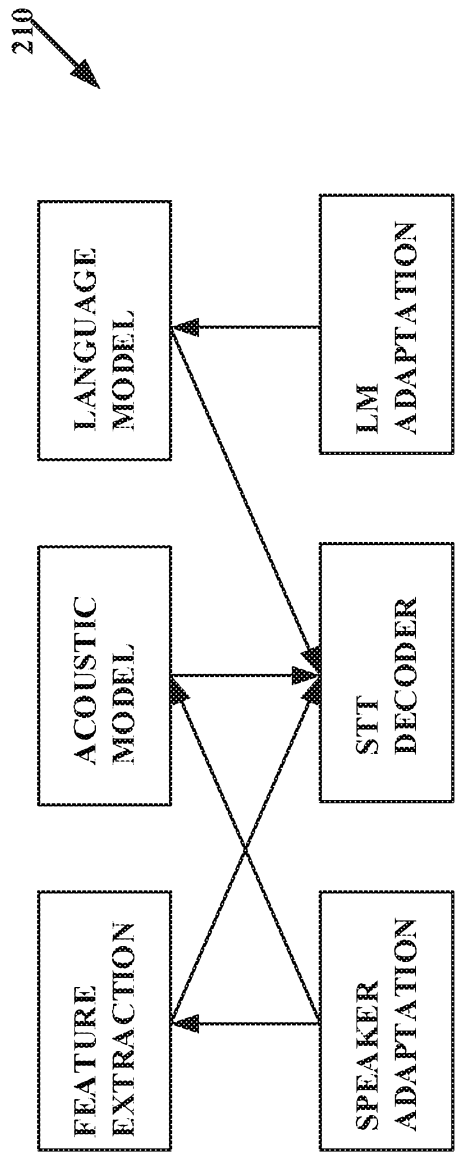
210
LANGUAGE MODEL
LM ADAPTATION
ACOUSTIC MODEL
STT DECODER
FEATURE EXTRACTION
SPEAKER ADAPTATION

FIG. 3A

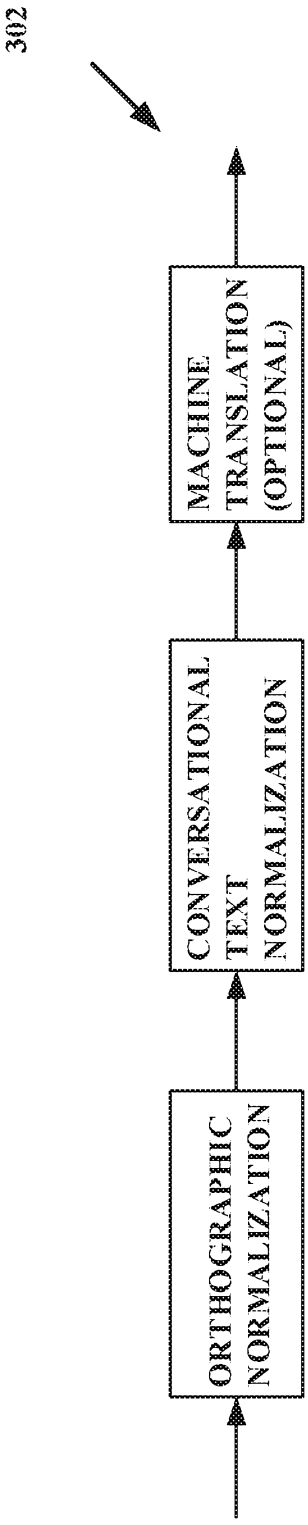
302
MACHINE TRANSLATION (OPTIONAL)
CONVERSATIONAL TEXT NORMALIZATION
ORTHOGRAPHIC NORMALIZATION

FIG. 3B

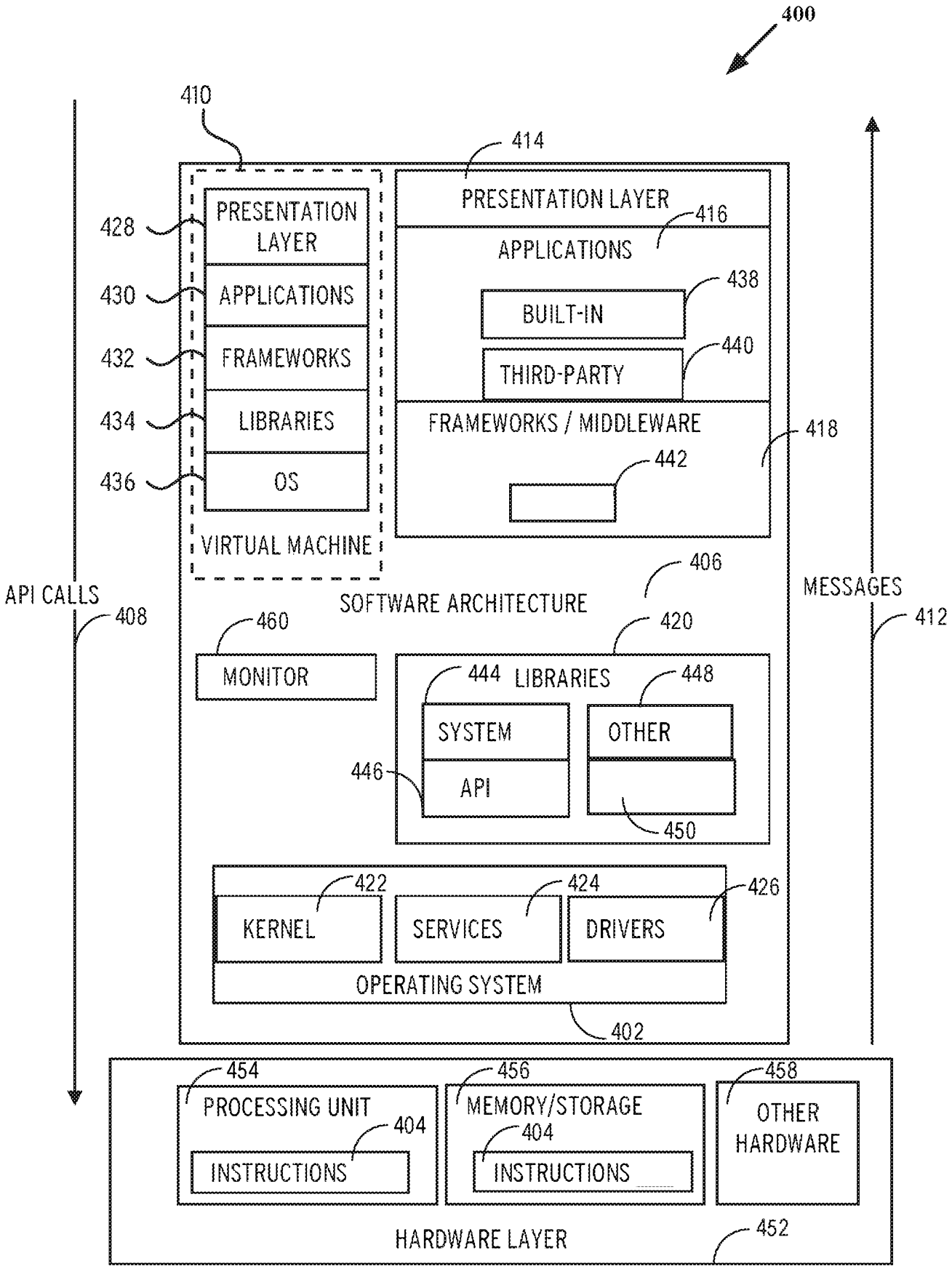
400
410
414
PRESENTATION LAYER
428
APPLICATIONS
430
FRAMEWORKS
432
LIBRARIES
434
OS
436
VIRTUAL MACHINE
PRESENTATION LAYER
416
APPLICATIONS
438
BUILT-IN
440
THIRD-PARTY
FRAMEWORKS / MIDDLEWARE
418
442
API CALLS
408
SOFTWARE ARCHITECTURE
406
MESSAGES
412
460
MONITOR
420
444
LIBRARIES
448
SYSTEM
OTHER
446
API
450
422
424
426
KERNEL
SERVICES
DRIVERS
OPERATING SYSTEM
402
454
456
458
PROCESSING UNIT
404
INSTRUCTIONS
MEMORY/STORAGE
404
INSTRUCTIONS
OTHER HARDWARE
HARDWARE LAYER
452

FIG. 4

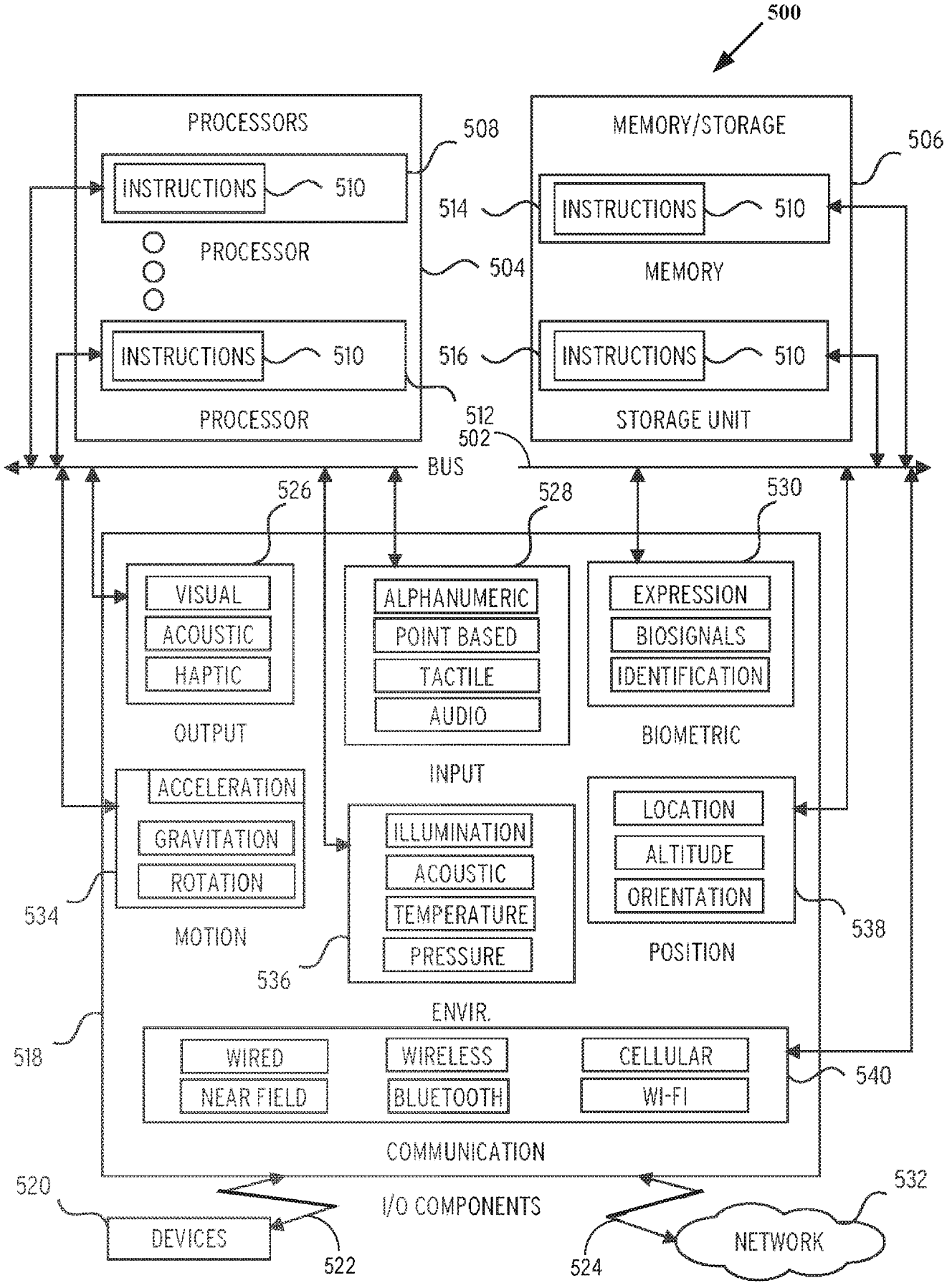
500
PROCESSORS
508
INSTRUCTIONS
510
PROCESSOR
504
INSTRUCTIONS
510
PROCESSOR
512
MEMORY/STORAGE
506
514
INSTRUCTIONS
510
MEMORY
516
INSTRUCTIONS
510
STORAGE UNIT
502
BUS
526
528
530
VISUAL
ACOUSTIC
HAPTIC
OUTPUT
ALPHANUMERIC
POINT BASED
TACTILE
AUDIO
INPUT
EXPRESSION
BIOSIGNALS
IDENTIFICATION
BIOMETRIC
ACCELERATION
GRAVITATION
ROTATION
534
MOTION
ILLUMINATION
ACOUSTIC
TEMPERATURE
PRESSURE
536
ENVIR.
LOCATION
ALTITUDE
ORIENTATION
538
POSITION
518
WIRED
NEAR FIELD
WIRELESS
BLUETOOTH
CELLULAR
WI-FI
540
COMMUNICATION
520
I/O COMPONENTS
DEVICES
522
524
NETWORK
532

FIG. 5

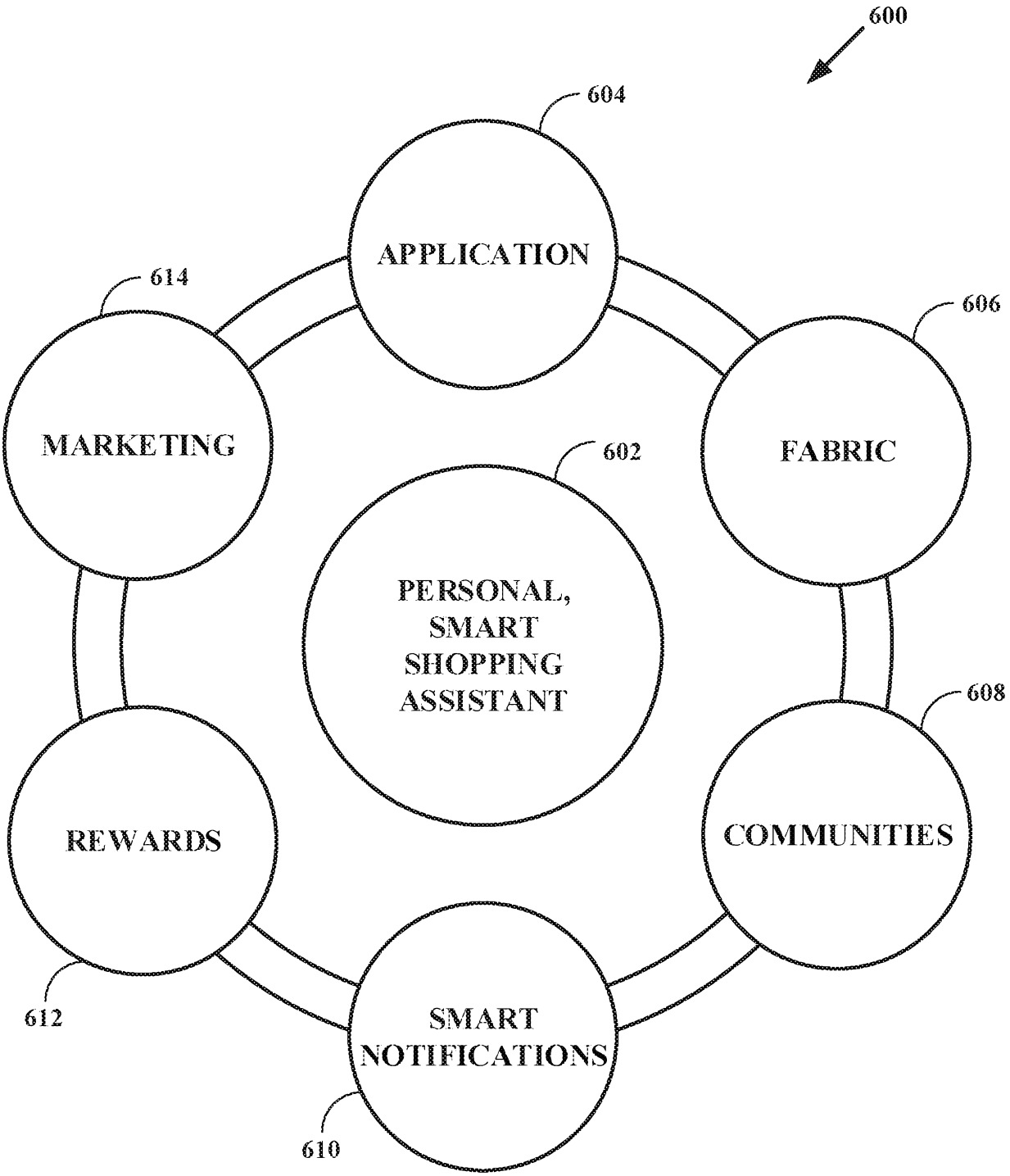
600
604
APPLICATION
614
MARKETING
606
FABRIC
602
PERSONAL, SMART SHOPPING ASSISTANT
608
REWARDS
COMMUNITIES
612
SMART NOTIFICATIONS
610

FIG. 6

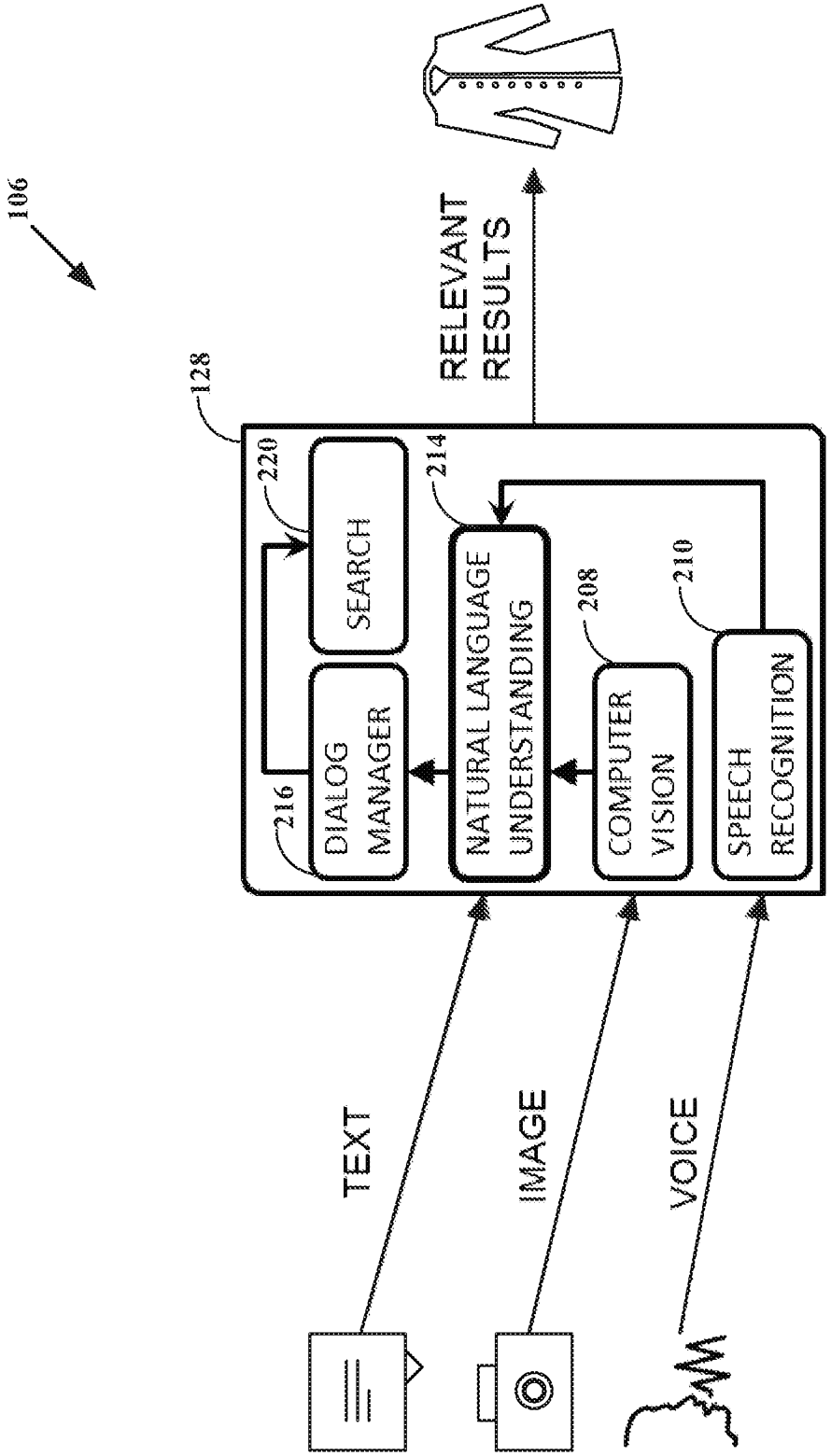
106
128
RELEVANT RESULTS
220
SEARCH
214
NATURAL LANGUAGE UNDERSTANDING
208
COMPUTER VISION
210
SPEECH RECOGNITION
216
DIALOG MANAGER
TEXT
IMAGE
VOICE

FIG. 7

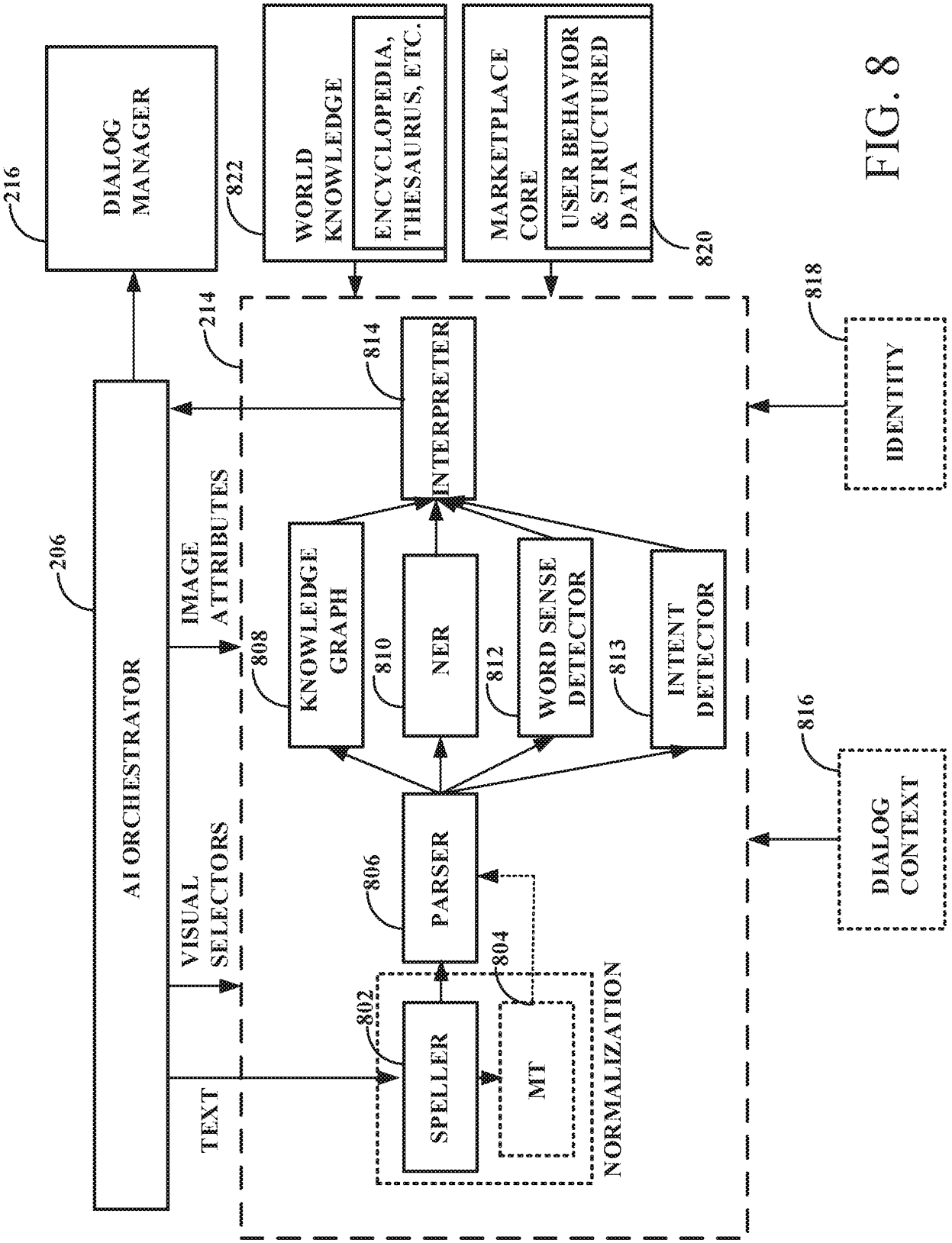
216
DIALOG MANAGER
822
WORLD KNOWLEDGE
ENCYCLOPEDIA, THESAURUS, ETC.
MARKETPLACE CORE
USER BEHAVIOR & STRUCTURED DATA
820
214
814
INTERPRETER
818
IDENTITY
206
AI ORCHESTRATOR
IMAGE ATTRIBUTES
808
KNOWLEDGE GRAPH
810
NER
812
WORD SENSE DETECTOR
813
INTENT DETECTOR
816
DIALOG CONTEXT
VISUAL SELECTORS
806
PARSER
804
802
SPELLER
MT
NORMALIZATION
TEXT

FIG. 8

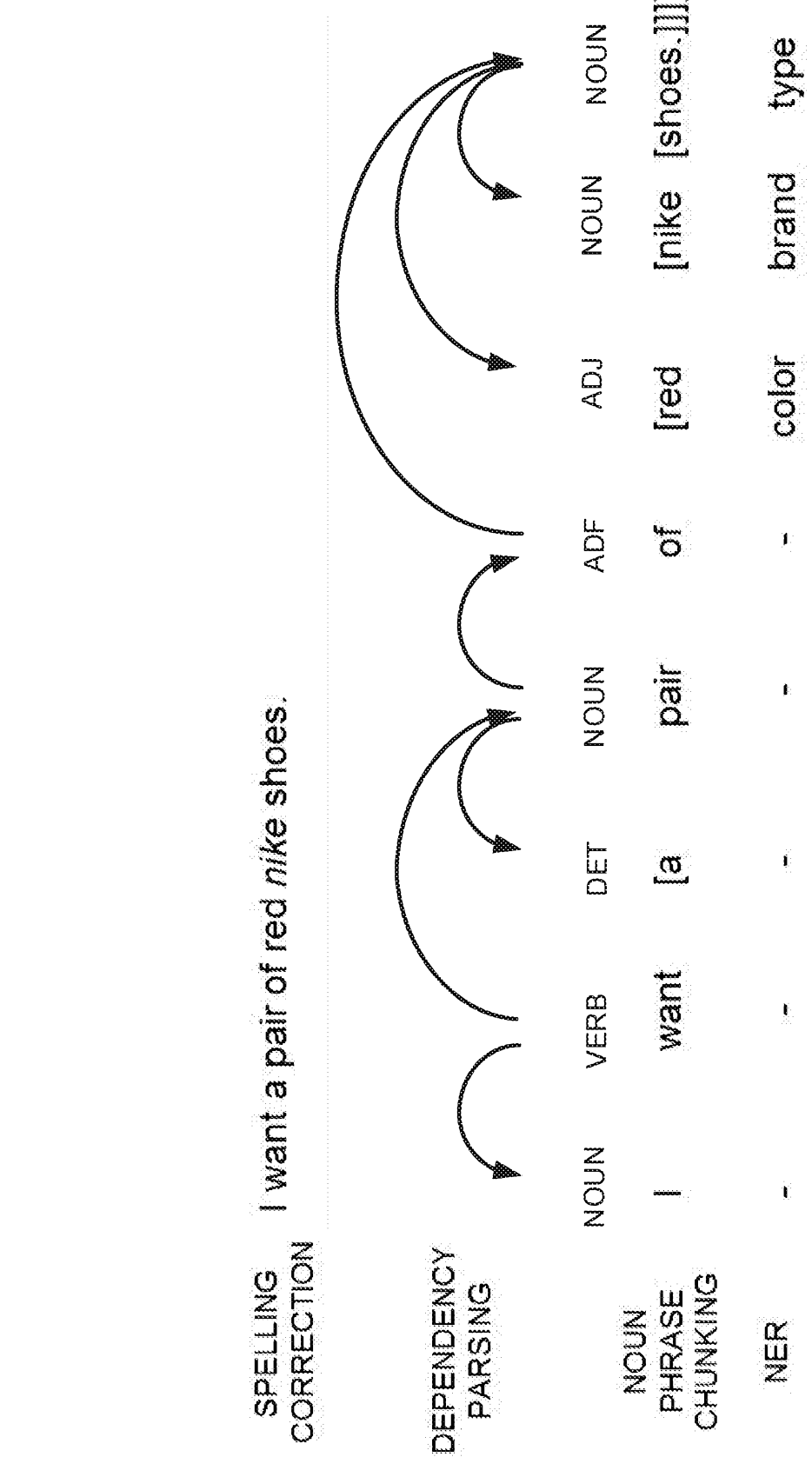
SPELLING CORRECTION
I want a pair of red nike shoes.
DEPENDENCY PARSING
NOUN VERB DET NOUN ADF ADJ NOUN NOUN
NOUN PHRASE CHUNKING
I want [a pair of [red [nike [shoes.]]]]
NER
- - - - - color brand type

FIG. 9

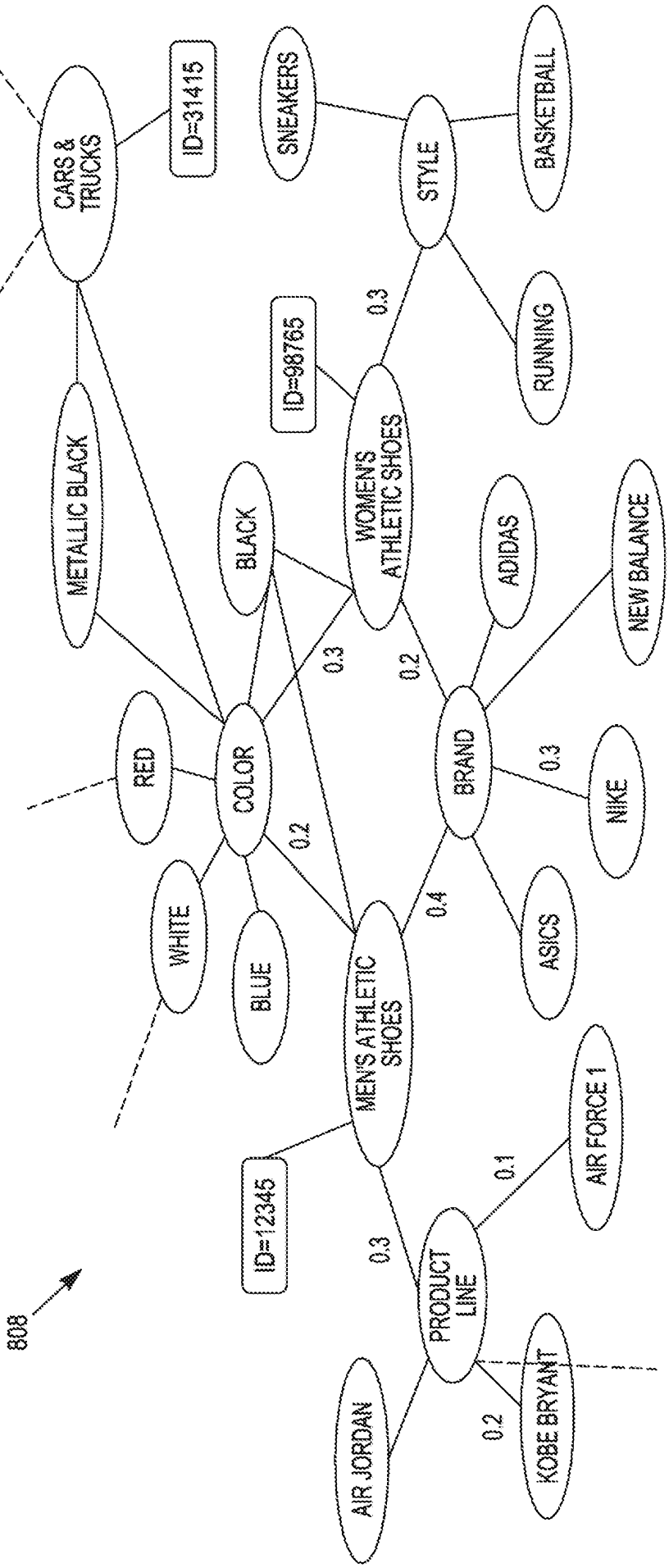
808
CARS & TRUCKS
ID=31415
SNEAKERS
STYLE
BASKETBALL
RUNNING
ID=98765
0.3
METALLIC BLACK
BLACK
WOMEN'S ATHLETIC SHOES
ADIDAS
NEW BALANCE
0.3
0.2
RED
COLOR
BRAND
0.3
NIKE
0.2
WHITE
BLUE
MEN'S ATHLETIC SHOES
0.4
ASICS
ID=12345
0.3
PRODUCT LINE
0.1
AIR FORCE 1
AIR JORDAN
0.2
KOBE BRYANT

FIG. 10

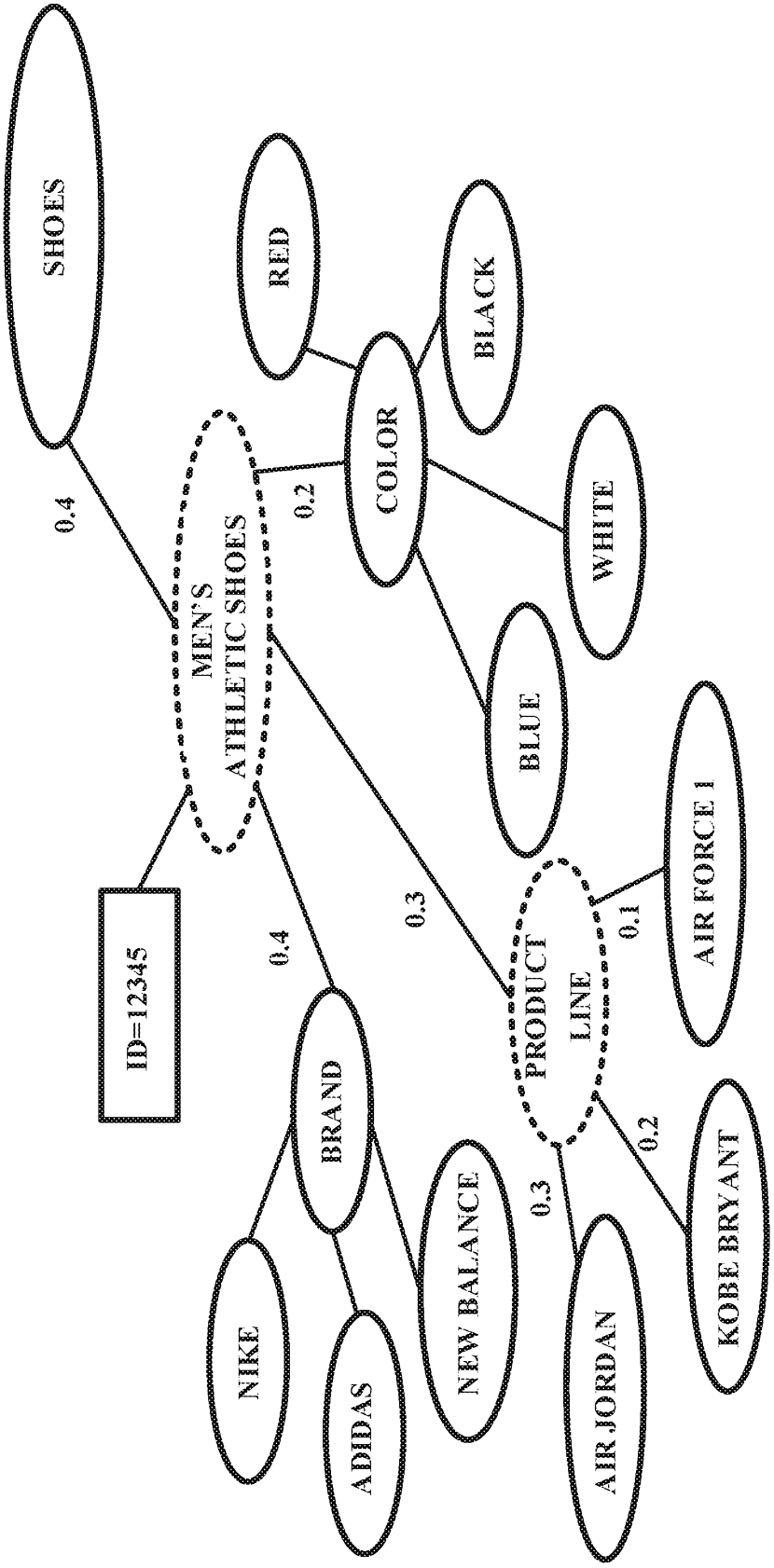
ID=12345
SHOES
MEN'S ATHLETIC SHOES
0.4
0.4
0.2
0.3
COLOR
RED
BLACK
WHITE
BLUE
BRAND
NIKE
ADIDAS
NEW BALANCE
PRODUCT LINE
0.1
0.3
0.2
AIR FORCE 1
AIR JORDAN
KOBE BRYANT

FIG. 11A

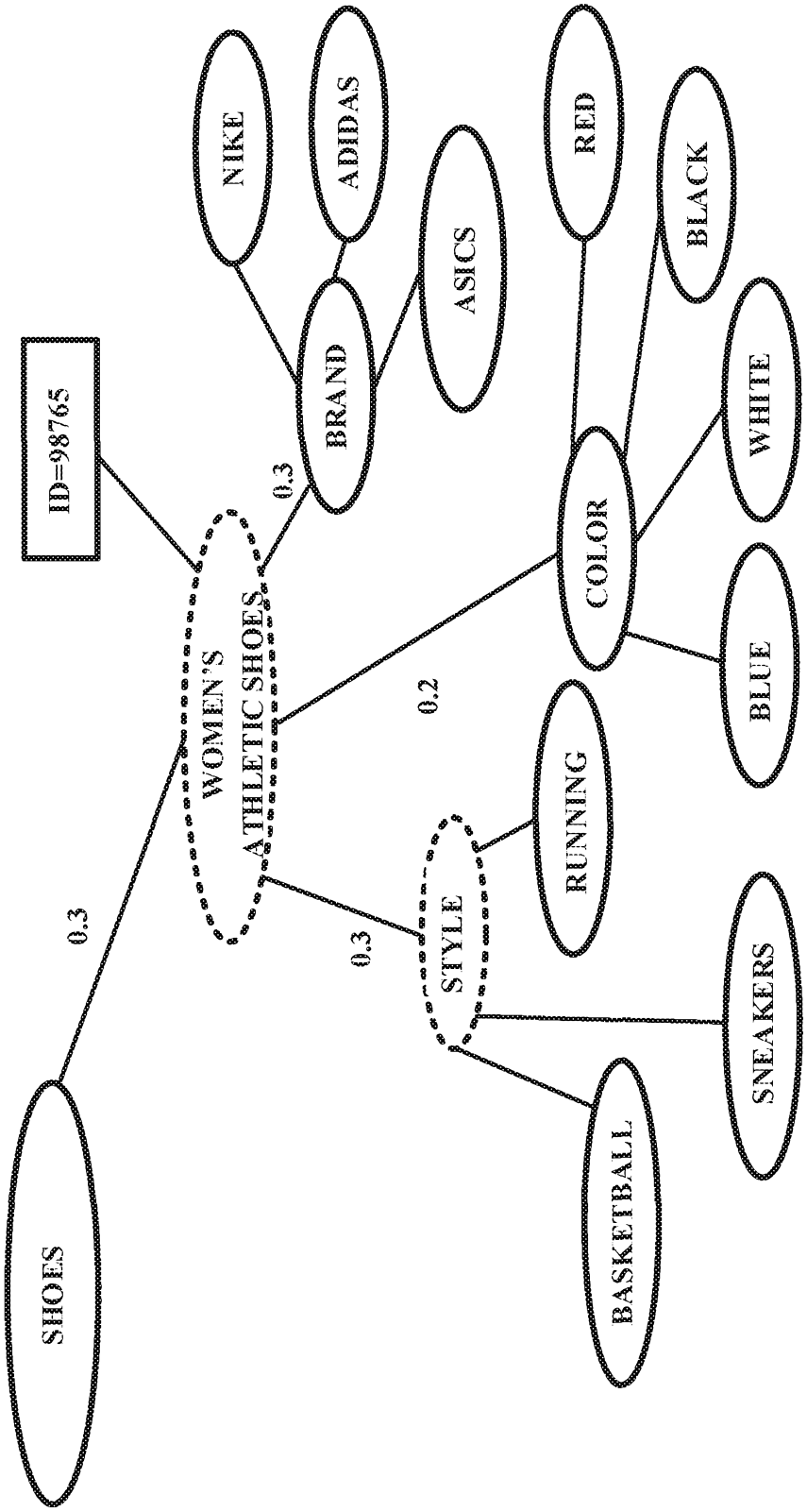
SHOES
0.3
ID=98765
WOMEN'S ATHLETIC SHOES
0.3
BRAND
NIKE
ADIDAS
ASICS
0.2
COLOR
RED
BLACK
WHITE
BLUE
0.3
STYLE
RUNNING
BASKETBALL
SNEAKERS

FIG. 11B

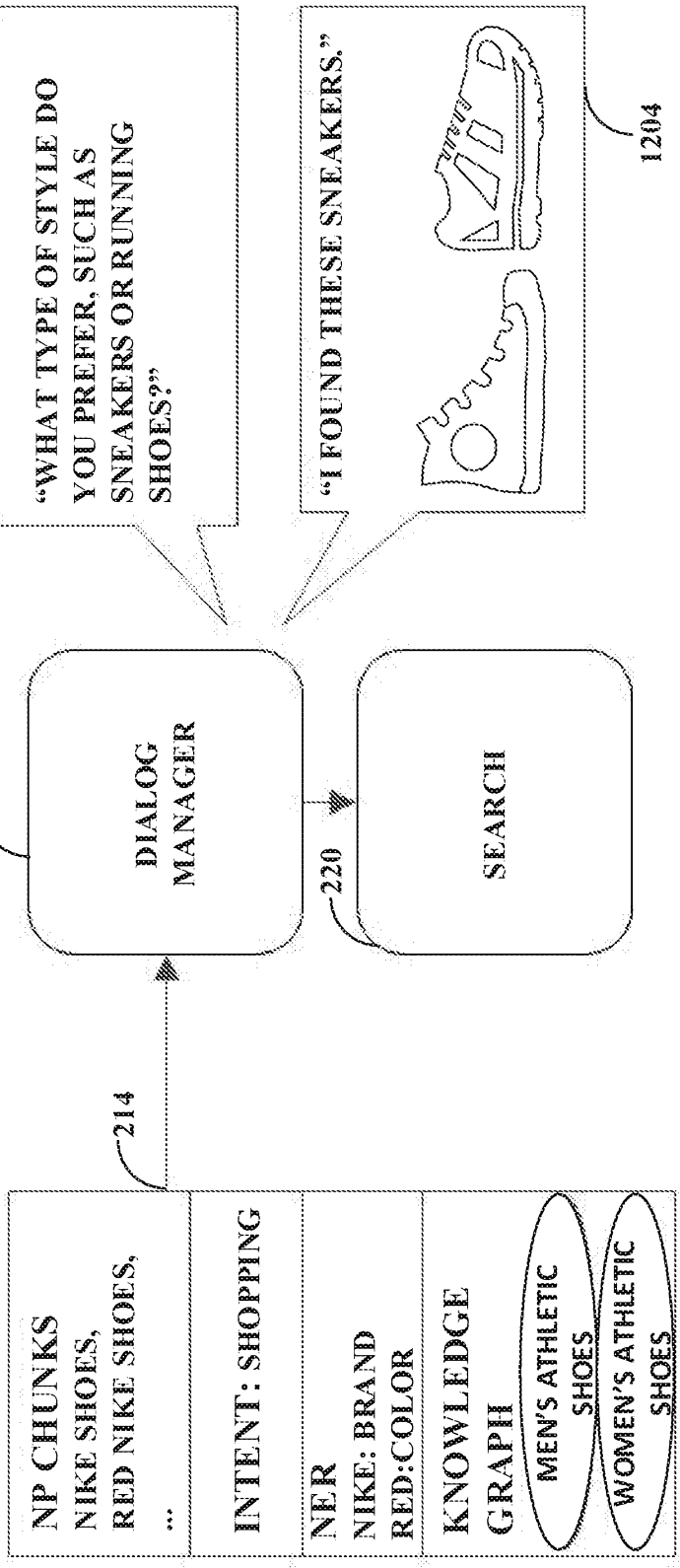
1202
"WHAT TYPE OF STYLE DO YOU PREFER, SUCH AS SNEAKERS OR RUNNING SHOES?"
"I FOUND THESE SNEAKERS."
1204
216
DIALOG MANAGER
220
SEARCH
214
NP CHUNKS
NIKE SHOES,
RED NIKE SHOES,
...
INTENT: SHOPPING
NER
NIKE: BRAND
RED:COLOR
KNOWLEDGE GRAPH
MEN'S ATHLETIC SHOES
WOMEN'S ATHLETIC SHOES

FIG. 12

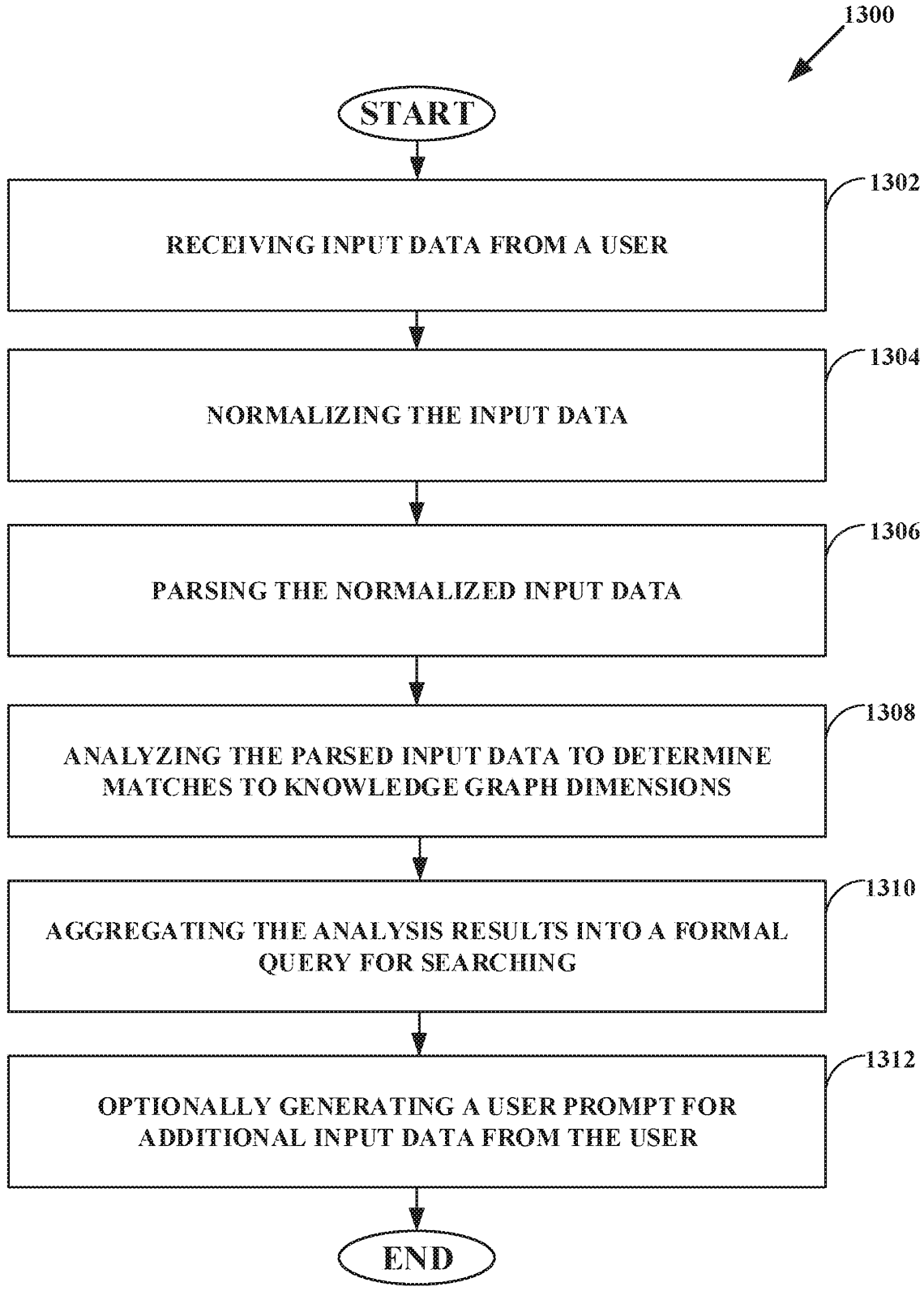
1300
START
RECEIVING INPUT DATA FROM A USER
1302
NORMALIZING THE INPUT DATA
1304
PARSING THE NORMALIZED INPUT DATA
1306
ANALYZING THE PARSED INPUT DATA TO DETERMINE MATCHES TO KNOWLEDGE GRAPH DIMENSIONS
1308
AGGREGATING THE ANALYSIS RESULTS INTO A FORMAL QUERY FOR SEARCHING
1310
OPTIONALLY GENERATING A USER PROMPT FOR ADDITIONAL INPUT DATA FROM THE USER
1312
END

FIG. 13