



- (51) **International Patent Classification:**
G06F 17/27 (2006.01)
- (21) **International Application Number:**
PCT/US20 17/0446 10
- (22) **International Filing Date:**
31 July 2017 (31.07.2017)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/368,975 29 July 2016 (29.07.2016) US
- (72) **Inventors; and**
- (71) **Applicants:** SWART, Erik [US/US]; 1110 112th Avenue NE, Suite 500, Bellevue, WA 98004 (US). DROUIN, Emilie [US/US]; 1110 112th Avenue NE, Suite 500, Bellevue, WA 98004 (US).
- (74) **Agent:** KOO, Hean, L. et al; Sheppard Mullin Richter & Hampton LLP, 2099 Pennsylvania Avenue, NW, Suite 100, Washington, DC 20006 (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,

(54) **Title:** SYSTEM AND METHOD OF DISAMBIGUATING NATURAL LANGUAGE PROCESSING REQUESTS

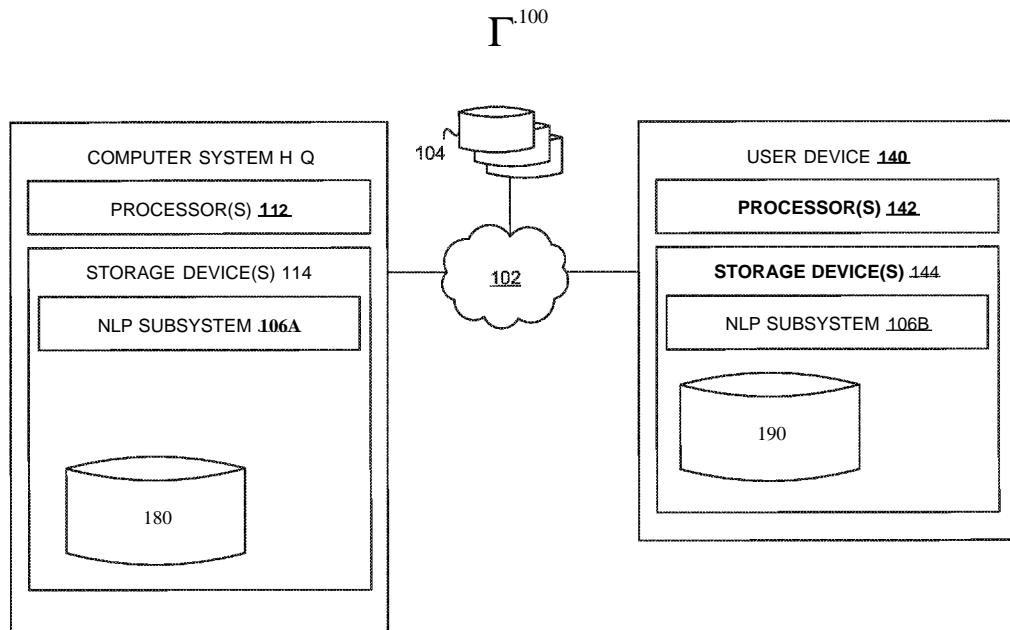


FIG. 1

(57) **Abstract:** A system and method is provided of disambiguating natural language processing requests based on smart matching, request confirmations that are used until ambiguities are resolved, and machine learning. Smart matching may match entities (e.g., contact names, place names, etc.) based on user information such as call logs, user preferences, etc. If multiple matches are found and disambiguation has not yet been learned by the system, the system may request that the user identify the intended entity. On the other hand, if disambiguation has been learned by the system, the system may execute the request without confirmations. The system may use a record of confirmations and/or other information to continuously learn a user's inputs in order to reduce ambiguities and no longer prompt for confirmations.



WO 2018/023106 A1

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

SYSTEM AND METHOD OF DISAMBIGUATING NATURAL LANGUAGE PROCESSING REQUESTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 62/368,975, entitled "SYSTEM AND METHOD OF DISAMBIGUATING NATURAL LANGUAGE PROCESSING REQUESTS", filed July 29, 2016, which is hereby incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

[002] The invention relates to a system and method of disambiguating natural language processing requests based on smart matching, request confirmations that are used until ambiguities are resolved, and machine learning.

BACKGROUND OF THE INVENTION

[003] Facilitating human to machine interactions in a natural manner is a difficult problem. Both non-voice and certain voice interfaces tend to be overly structured and require some level of familiarity with the interfaces. An ability for a human to interact with machines using natural language utterances in various contexts remains a desirable goal. However, human interaction with machines in various contexts remains a difficult problem.

[004] One area of difficulty lies in the disambiguation of natural language requests. The objects of natural language requests and command, entity names, may sometimes be ambiguous. For example, a user may have multiple contacts of the same name in their directory. A phone dialing request made by first name may require disambiguation. A natural language system however, that constantly requests a user to use specific and unambiguous language or to clarify requests made in ambiguous language may quickly begin to feel burdensome for the user.

[005] These and other drawbacks exist with conventional natural language processing disambiguation methods.

SUMMARY OF THE INVENTION

[006] The invention addressing these and other drawbacks relates to a system and method of machine learned disambiguation of natural language processing requests. An intent of a natural language input (e.g., a natural language utterance of a user) may be ambiguous. For example, a user may provide the input "call John," when there are several "Johns" in the user's contact list. Other types of ambiguities and other types of requests may be ambiguous based on natural language processing as well.

[007] The system may disambiguate natural language processing requests based on smart matching, request confirmations that are used until ambiguities are resolved, and machine learning. Smart matching may match entities (e.g., contact names, place names, etc.) based on user information such as call logs, user preferences, etc. If multiple matches are found and disambiguation has not yet been learned by the system for these matches, the system may request that the user identify the intended entity. On the other hand, if disambiguation has been learned by the system, the system may execute the request without confirmations. The system may use a record of confirmations and/or other information to continuously learn a user's inputs in order to reduce ambiguities and no longer prompt for confirmations.

[008] To this end, the system may include a smart matching engine that identifies and matches entities, a confirmation engine that generates request confirmations (e.g., "do you want to call John Doe or John Smith?") that are used until a sufficient level of confidence that an ambiguity relating to a given natural language input has been resolved (e.g., until the system is confident that the user intends to call "John Doe" and not "John Smith"), a learning engine that uses machine learning based on user information (e.g., history of request confirmations from a user, where each request confirmation sought to resolve a prior ambiguous result of natural language processing from a prior natural language input), and/or other components to disambiguate a natural language input.

[009] The smart matching engine may identify an input entity name. The smart matching engine may then match the input entity name to an entity name that can be acted upon. For example, if the user requests directions to Main St., then the smart matching engine may operate to suggest a match between the user uttered input entity name "Main St." and an identified entity name - i.e., an actual, mappable, Main St. If the user asks to "call James," then the smart matching engine may operate to suggest a match between an identified entity name, i.e., an actual contact in the user's directory, and the input entity name of "James."

[01 0] The smart matching engine may suggest a match between an input entity name and an identified entity name based at least partially on an entity matching model. The entity matching model may comprise information compiled through machine learning techniques, and may include at least context information and user history information, as well as other information suitable for machine learning. A suggested match may include the input entity name and an identified entity name. A suggested match may further include additional entity match information.

[01 1] Additional entity match information may include context information related to the natural language request, the input entity name, and the identified entity name. For example, for a user that lives in New York, an entity matching model may associate the entity name "taxi" in the request "call a taxi" with a particular taxi company in New York. When creating a suggested match for the input entity name "taxi," for that user, the smart matching engine may include the additional entity match information of the user's location as part of the suggested match. This may permit the system to disambiguate the term "taxi" differently based on the user's location.

[01 2] Context information usable by the smart matching engine may include a request context. For example, if a user makes a request to "call James" during the business day, the smart matching engine may suggest a match between the input entity name James and a business contact named James. If the request is made after hours, the smart matching engine may suggest a match with a contact from a friends and family contact list. Similarly, when attempting to disambiguate requests for directions, smart matching engine may suggest matches between locations that are closer to a present location of the user.

[01 3] User history information usable by the smart matching engine may include information about previous user requests and activity. For example, when requesting to make a call, the smart matching engine may select a name from a user's contact list that the user calls very frequently. When seeking directions, the smart matching engine may suggest a destination location that matches a location the user has travelled to frequently.

[014] The smart matching engine may additionally employ machine learned matching techniques. For example, a user may frequently refer to a contact name by a nickname, making a request to "call Jim," where the name "James" is stored in the user's contact list. Such a connection may become a part of an entity matching model based on machine learning techniques, as described in more detail below.

[01 5] In some implementations, the smart matching engine may suggest a match between an

input entity name and an identified entity name based on an exact match. For example, where a user requests to "call John," the system may suggest only matches that match "John" exactly. In some implementations, the smart matching engine may suggest a match between an input entity name and an identified entity name based on a phonetic match. For example, where a user requests to "call John," the system may suggest all matches that sound like "John," including, for example, "Jon." In some implementations, the smart matching engine may suggest a match between an input entity name and an identified entity name based on similar sounds. Continuing to use the above example, the smart matching engine may suggest a match between "John" and "Juwana." In some implementations the smart matching engine may suggest a match between an input entity name and an identified entity name based on an interpretation of a user's pronunciation style.

[016] In some implementations, the smart matching engine may suggest a match between an input entity name and an identified entity name based on learned associations. For example, a user may request to "call my brother." The smart matching engine may suggest a match between the input entity name "my brother," and all user contacts having a same last name as the user. The smart matching engine may also request clarification from the user about the entity name "my brother." Once the system has successfully confirmed a match between a user contact and the input entity name "my brother," the entity matching model may be updated with the information in order to correctly link the input entity name "my brother" to the appropriate contact. In another example, a user may request directions to "the pizza place." Smart matching engine may propose a suggested match between the input entity name "the pizza place," and the closest pizza restaurant. Once the system has successfully confirmed a match between "the pizza place," and a particular pizza restaurant, the entity matching model may be updated to associate "the pizza place," with the specific pizza restaurant.

[017] In some implementations, the smart matching engine may pass a suggested match between an input named entity and an identified named entity to confirmation engine for command confirmation, as described below.

[018] In some implementations, the smart matching engine may determine an alternate identified entity name if the confirmation engine is unable to confirm the suggested match between the input named entity and the identified entity name. In some implementations, the smart matching engine may determine an alternate identified entity name using similar methods to those described above for selecting the first identified entity name. In determining an alternate identified entity name, the smart matching engine may select a next best match for an

input entity name. In some implementations, the smart matching engine may develop an n-best list of suggested matches for an input entity name including n number of potential matches ranked in order of best fit. The smart matching engine may provide the 2nd, 3rd, 4th, etc., matches from an n-best list, in order, to the confirmation engine as alternate identified names if the first suggestion is cancelled or met with non-confirmation. In some implementations, the smart matching engine may provide an entire n-best list to the confirmation engine.

[019] The confirmation engine may perform command confirmation based on a user's confirmation history of a proposed entity match. Command confirmation may be performed actively and/or passively. For example, active command confirmation may include querying the user to confirm a suggest match - e.g., where a user has asked to "call James," a command confirmation may query the user "did you mean James Smith?" In some implementations, passive command confirmation may include permitting a user a predefined amount of time to cancel an action. For example, where a user has asked to "call James," a command confirmation may alert the user "calling James Smith," and wait for a predetermined amount of time to give the user a chance to cancel the action before executing the action. If the user does not cancel the action, then the suggested entity match may be confirmed. A predetermined amount of time may be relatively short, e.g., 1 to 5 seconds, or relatively long, e.g., 5-15 seconds. Selecting which command confirmation method to use may be based on a user's confirmation history.

[020] For example, the confirmation engine may use a history of a user's confirmation between a particular input entity name and a particular identified entity name. For example, a user may ask to "call James," and the smart matching engine may determine a suggested match between the input entity name "James" and an identified entity name "James Smith," corresponding to a user contact. If the smart matching engine has previously suggested the same match, and the user has previously provided confirmations, the confirmation engine may alert the user of the impending action. If the user has not previously provided enough confirmations of the suggested match, the confirmation engine may request active confirmation from the user. The confirmation engine may switch between these two methods when a number of previous confirmations of the suggested match exceeds a predetermined threshold number of times. Such a predetermined threshold number may be one or more times. A predetermined threshold number may vary depending on the type of entity name that is being matched (e.g., whether it is a contact name or a location address.)

[021] In some implementations, the confirmation engine may update a number of previous

confirmations of a suggested match in a user's confirmation history after confirmation or non-confirmation is received from the user. If confirmation is received from the user for a particular suggested match, the confirmation engine may increment the number of previous confirmations. If a user cancels or fails to confirm a suggested match, the confirmation engine may instead decrement the number of previous confirmations. In some implementations, the confirmation engine may reset a number of previous confirmations to zero after a non-confirmation is received. In some implementations, the confirmation engine may separately record a number of non-confirmations and decrement or reset the previous confirmation score after a threshold number of non-confirmations is reached. In some implementations, the confirmation engine may separately record a number of consecutive non-confirmations and decrement or reset the previous confirmation score after a threshold number of consecutive non-confirmations is reached. In some implementations, the confirmation engine may do nothing to a previous confirmation score when a non-confirmation is received.

[022] In some implementations, the confirmation engine may request an alternate identified match from the smart matching engine if a non-confirmation or cancellation is received. The smart matching engine may provide a next best alternate identified match to the confirmation engine. The confirmation engine may treat the alternate identified match similarly to the initial identified match, and determine an active or passive confirmation method based on a number of previous confirmations.

[023] In some implementations, the confirmation engine may receive an n-best list from the smart matching engine. The confirmation engine may then try to confirm with the user each entry in the n-best list as a selected match. In some implementations, the confirmation engine may try each member of the n-best list in turn, using the above-discussed confirmation methods based on confirmation history. In some implementations, the confirmation engine may try to confirm the first entry of an n-best list, and, if confirmation fails, query the user to select from the remaining entries in the n-best list.

[024] The learning engine may provide machine learning updates to a smart matching model used by the smart matching engine and confirmation engine in suggesting and confirming matches between natural language inputs and named entities, and/or other instructions that computer system 110 to perform various operations, each of which are described in greater detail herein.

[025] The learning engine may include instructions for updating an entity matching model based on results provided by the smart matching engine and the confirmation engine. As

described above, the smart matching engine may suggest a match between an input entity name and an identified entity name, while the confirmation engine may operate to confirm the identified entity name and determine it as a selected entity name. When an entity name is confirmed as a selected entity name, the learning engine may update the entity matching model with information about the confirmed match. The learning engine may update context information of the entity matching model with information about the context (e.g., time, location, whether the user is traveling, etc.) under which the suggested match was confirmed. The learning engine may update user history information of the entity matching model, for example, to keep track of a frequency of user requests involving the selected entity name.

[026] For example, where a user has a requested to make a phone call or send a text message to a contact and the confirmation engine confirms the suggested match provided by the smart matching engine, the learning engine may update the entity matching model with information about the match between the input entity name and the identified entity name, including, for example, time of day, location of the user, category of the entity name (e.g., business or personal contact), and any other suitable information. In some implementations, when the confirmation engine obtains a non-confirm result of a suggested match, the learning engine may update entity matching model with information that the match was not confirmed, as well as other context information associated with the failed match.

[027] For example, a user may frequently his brother "James Lewis" in the evening but call his business associate "James Smith" during the day. Due to a frequency of calls, the smart matching engine may initially suggest a match between the request to "call James," and "James Smith," even when the call is taking place after 9 pm. After one or more non-confirmations between a command to "call James" at 9 pm and "James Smith," the learning engine may sufficiently update the entity matching model to make an association between "James" and the user's brother "James Lewis" in the evening, and between "James" and the user's business associate "James Smith" during the day.

[028] These and other objects, features, and characteristics of the system and/or method disclosed herein, as well as the methods of operation and functions of the related elements of structure and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and

description only and are not intended as a definition of the limits of the invention. As used in the specification and in the claims, the singular form of "a", "an", and "the" include plural referents unless the context clearly dictates otherwise.

BRIEF DESCRIPTION OF THE DRAWINGS

[029] FIG. 1 illustrates a system for natural language processing, according to an implementation of the invention.

[030] FIG. 2 depicts an NLP system configured to perform natural language processing on an utterance, according to an implementation of the invention.

[031] FIG. 3 depicts a process of natural language processing disambiguation, according to an implementation of the invention.

[032] Fig. 4 depicts a process of disambiguation machine learning, according to an implementation of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[033] The invention described herein relates to a system and method of natural language processing disambiguation. In particular, the systems and methods described are related to machine learning of disambiguation models. The system may be configured to receive and process natural language inputs, and then use machine learning techniques to optimally provide disambiguation between potentially ambiguous inputs.

[034] For example, a user may have multiple contacts having the same first or last name in a directory. If a user makes a natural language request to "Call John," when there are three Johns stored in the users contact directory, the system may employ disambiguation models to determine which John the user the intended. The system may review context data, including call frequency, call timing, user location, and other information, to determine a suggestion to match the user's request. The user may then be prompted to confirm the suggestion, and/or, in some implementations, be provided with a time in which to cancel the suggestion, before the request is carried out using the suggestion. The system may incorporate the user response in disambiguation models to optimize future suggestions and actions. Further examples of user requests requiring disambiguation may include requests for directions (e.g., where multiple destinations may have similar names), requests for weather reports (e.g., where the user fails to specify the location of request), requests to play music (e.g., where the user selects an album but not an artist) and others. Disambiguation may be required between identical user requests,

between phonetically similar requests, between homophonic requests, and/or between requests that may sound similar based on user pronunciation. The system may further be configured for disambiguation of generic requests to identify a user's specific intent. For example, a request to "call the pizza place," may be disambiguated to a phone dialing request to contact a specific pizza restaurant that a user frequently calls.

[035] FIG. 1 illustrates a system 100 for natural language processing. In one implementation, system 100 may include a computer system 110, a user device 140, a database 104, and/or other components. Components of system 100 may communicate with one another via network 102. Network 102 may be any type of network, such as a WAN, LAN, Bluetooth network, the Internet, etc.

[036] Computer system 110 may be configured as a server (e.g., having one or more server blades, processors, etc.), a gaming console, a handheld gaming device, a personal computer (e.g., a desktop computer, a laptop computer, etc.), a smartphone, a tablet computing device, and/or other device that can be programmed for natural language processing, and, in particular the disambiguation of natural language processing requests.

[037] Computer system 110 may include one or more processors 112 (also interchangeably referred to herein as processors 112, processor(s) 112, or processor 112 for convenience), one or more storage devices 114, and/or other components. Processors 112 may be programmed by one or more computer program instructions. For example, processors 112 may be programmed by NLP subsystem 106A, and/or other instructions that program computer system 110 to perform various operations, each of which are described in greater detail herein. As used herein, for convenience, the various instructions and engines will be described as performing an operation, when, in fact, the various instructions program the processors 112 (and therefore computer system 110) to perform the operation. Storage device 114 may further include a user history database 180.

[038] NLP subsystem 106A may perform at least some aspects of natural language processing on an utterance using all or a portion of the components of NLP subsystem 106 illustrated in FIG. 2. In other words, computer system 110 may be programmed with some or all of the functions of NLP system 106 described with respect to FIG. 2.

[039] User device 140 may be configured as a server device, a gaming console, a handheld gaming device, a personal computer (e.g., a desktop computer, a laptop computer, etc.), a smartphone, a tablet computing device, and/or other device that can be programmed to receive and interpret natural language processing requests. In particular, user device 140 may be

programmed to perform disambiguation on a received natural language input.

[040] User device 140 may include one or more processors 142 (also interchangeably referred to herein as processors 142, processor(s) 142, or processor 142 for convenience), one or more storage devices 144, and/or other components. Processors 142 may be programmed by one or more computer program instructions. For example, processors 142 may be programmed by NLP subsystem 106B, and/or other instructions that program computer system 110 to perform various operations, each of which are described in greater detail herein. As used herein, for convenience, the various instructions will be described as performing an operation, when, in fact, the various instructions program the processors 142 (and therefore user device 140) to perform the operation. Additionally storage device 144 may include one or more databases, including a user information a user history database 190.

[041] Any or all aspects of natural language processing may be carried out by programming instructions instantiated on either NLP subsystem 106b of user device 140 and NLP subsystem 106a of computer system 110. Each of user device 140 and computer system 110 may perform all or a portion of a natural language processing task. User device 140 and computer system 110 may operate to perform natural language processing tasks sequentially, communicating results via network 102. In such implementations, one system may pass results of a completed task to the other system for further processing. User device 140 and computer system may operate to redundantly perform natural language processing tasks. In such implementations, depending upon the results of a natural language processing task carried out by one system, the other system may repeat the same task. User device 140 and computer system 110 may operate in parallel to perform natural language processing tasks. In such implementations, user device 140 and computer system 110 may each perform one or more tasks independently and, after task completion, communicate with each other to select between and or combine the processing results.

[042] NLP subsystem 106B may perform at least some aspects of natural language processing on an utterance using all or a portion of the components of NLP subsystem 106 illustrated in FIG. 2. In other words, user device 140 may be programmed with some or all of the functions of NLP system 106 described with respect to FIG. 2.

[043] Although illustrated in FIG. 1 as a single component, computer system 110 and user device 140 may each include a plurality of individual components (e.g., computer devices) each programmed with at least some of the functions described herein. In this manner, some components of computer system 110 and/or user device 140 may perform some functions while

other components may perform other functions, as would be appreciated. The one or more processors 112 may each include one or more physical processors that are programmed by computer program instructions. The various instructions described herein are exemplary only. Other configurations and numbers of instructions may be used, so long as the processor(s) 112 are programmed to perform the functions described herein.

[044] Furthermore, it should be appreciated that although the various instructions are illustrated in FIG. 1 as being co-located within a single processing unit, in implementations in which processor(s) 112 includes multiple processing units, one or more instructions may be executed remotely from the other instructions.

[045] The description of the functionality provided by the different instructions described herein is for illustrative purposes, and is not intended to be limiting, as any of instructions may provide more or less functionality than is described. For example, one or more of the instructions may be eliminated, and some or all of its functionality may be provided by other ones of the instructions. As another example, processor(s) 112 may be programmed by one or more additional instructions that may perform some or all of the functionality attributed herein to one of the instructions.

[046] The various instructions described herein may be stored in a storage device 114, which may comprise random access memory (RAM), read only memory (ROM), and/or other memory. The storage device may store the computer program instructions (e.g., the aforementioned instructions) to be executed by processor 112 as well as data that may be manipulated by processor 112. The storage device may comprise floppy disks, hard disks, optical disks, tapes, or other storage media for storing computer-executable instructions and/or data.

[047] The various components illustrated in FIG. 1 may be coupled to at least one other component via a network 102, which may include any one or more of, for instance, the Internet, an intranet, a PAN (Personal Area Network), a LAN (Local Area Network), a WAN (Wide Area Network), a SAN (Storage Area Network), a MAN (Metropolitan Area Network), a wireless network, a cellular communications network, a Public Switched Telephone Network, and/or other network. In FIG. 1, as well as in other drawing Figures, different numbers of entities than those depicted may be used. Furthermore, according to various implementations, the components described herein may be implemented in hardware and/or software that configure hardware.

[048] The various databases 104 described herein may be, include, or interface to, for

example, an Oracle™ relational database sold commercially by Oracle Corporation. Other databases, such as Informix™, DB2 (Database 2) or other data storage, including file-based, or query formats, platforms, or resources such as OLAP (On Line Analytical Processing), SQL (Structured Query Language), a SAN (storage area network), Microsoft Access™ or others may also be used, incorporated, or accessed. The database may comprise one or more such databases that reside in one or more physical devices and in one or more physical locations. The database may store a plurality of types of data and/or files and associated data or file descriptions, administrative information, or any other data.

[049] Natural Language Processing

[050] FIG. 2 depicts an NLP system 106 configured to perform natural language processing on an utterance, according to an implementation of the invention. NLP system 106 may include a an Automated Speech Recognition ("ASR") engine 210, an Intent Recognition Engine ("IRE") 220, text-to-speech engine 230, a dictionary 240, a user profile 250, an agent system 260, smart matching engine 270, confirmation engine 271, learning engine 272, execution engine 273, and/or other components. The components of NLP system 106 may communicate with one another via communication manager 202, which may mediate interactions between components of NLP system 106.

[051] NLP system 106 may process an utterance encoded into audio to determine words or phrases from the utterance. Alternatively, NLP system 106 may process text that was already detected from the utterance. In whichever manner the utterance is received (e.g., whether as encoded audio or text), NLP system 106 may determine an intent of the utterance using the words or phrases, formulate requests based on its intent, and facilitate generating a response to or executing the request.

[052] For example, in implementations in which an utterance is received as audio, ASR engine 210 may use words and phrases from dictionary 240 and information received from agent system 260 (discussed below) to recognize words from the utterance (i.e., the audio). The recognized words and phrases are processed by IRE 220, which determines the intent of the utterance and generates a request that corresponds to the utterance based on context, and information from agent system 260. The request may include a query (e.g., "what is the weather like in Seattle?") a command (e.g., "turn up the volume"), and/or other information.

[053] To process the intent of the utterance, IRE 220 may use keywords within the utterance itself to identify context. For example, for the utterance "what is the weather like in Seattle," IRE 220 may determine that the context is weather and a subject is "Seattle." In some

instances, the contextual information may be based on immediately prior utterances. In these instances, IRE 220 may generate a stack that stores (e.g., in Random Access Memory) information relating to one or more previous utterances in a given session (which may be defined by utterances received within a threshold time of one another). For example, a first utterance "what is the weather like in Seattle" may be followed up with a second utterance "book me a flight there." By itself, the second utterance "book me a flight there" may be non-specific because "there" is undefined in the second utterance. However, when processed as part of a stack that includes information relating to the first utterance "what is the weather like in Seattle," IRE 220 determines that "there" refers to "Seattle."

[054] A natural language input may comprise a request and an entity name. A request may include several types of request, including a query, an executable command, and/or any other actionable statement. For example, a request may include a command to place a phone call or send a text message, may include a query for directions or a location, may include a question about the weather, and/or may include a question for a search engine. Numerous additional examples of requests may exist and may be processed from a received natural language input. Entity names may include objects of the processed request. For example, input entity names may include a contact to be called or texted, an address or establishment for which directions are sought, a location or time for which the weather is requested, and/or a target of a search engine query. Numerous other examples exist. Thus, a request and an input entity name, together may form a complete natural language input request, including both an action to be taken an object subject to that action. IRE 220 may process a received utterance to determine at least a type of request and an entity name.

[055] Alternatively or additionally, NLP system 106 may use agent system 260, which includes system and domain specific agents, that assists in ASR, intent recognition, request generation, and/or request response. Agent system 260 may use nonvolatile storage (e.g., storage device 114 or storage device 144) to store data, parameters, history information, and locally stored content provided in the system databases 104 or other data sources. User specific data, parameters, and session and history information that may determine the behavior of the agents are stored in one or more user profiles 250. For example, user specific speech patterns may assist in ASR, while user preferences and history may assist in contextual determinations and, therefore, intent recognition.

[056] The system agent provides default functionality and basic services. A domain specific agent relates to a corresponding domain of information (e.g., a business listing domain may

relate to business listings, a contacts domain may relate to user contacts, a shopping domain may relate to commerce, etc.).

[057] Agent system 260 create queries to local databases or through data sources on the Internet or other networks. Commands typically result in actions taken by the device on which NLP system 106 operates, or to a remote device or data source.

[058] In implementations in which a response to a request is desired, agent system 260 may create a response string for presentation to the user. The response string may be presented as text and/or as speech, in which case the string is sent to the text to speech engine 230 to be output by a speaker (whether an on-board or off-board speaker).

[059] Text-to-speech engine 230 may convert text to speech, such as when information from NLP system 106 (e.g., responses to spoken queries) is to be presented to a user in speech format. Conventional text-to-speech processors may be used for such text-to-speech generation.

[060] A more detailed description of NLP processing is described in U.S. Patent Application Serial No. 10/452,147, entitled "Systems and Methods for Responding Natural Language Speech Utterance," filed on June 3, 2003, the disclosure of which is incorporated in its entirety herein. NLP system 260 may use the foregoing and/or other types of NLP processing techniques and systems.

[061] Smart matching engine 270 may include additional instructions that program computer system 110. The instructions of smart matching engine 270 may include, without limitation, instructions for suggesting named entities to appropriate portions of a received natural language input, and/or other instructions that program computer system 110 to perform various operations, each of which are described in greater detail herein.

[062] Smart matching engine 270 may operate to identify an input entity name. ASR engine 210 and IRE 220 may process a natural language input to determine a type of request and an input entity name. Smart matching engine 270 may then match the input entity name to an entity name that can be acted upon. For example, if the user requests directions to Main St., then smart matching engine 270 may operate to suggest a match between the user uttered input entity name "Main St." and an identified entity name - i.e., an actual, mappable, Main St. If the user asks to "call James," then smart matching engine 270 may operate to suggest a match between an identified entity name, i.e., an actual contact in the user's directory, and the input entity name of "James."

[063] Smart matching engine 270 may suggest a match between an input entity name and at

least one identified entity name based at least partially on an entity matching model. The entity matching model may comprise information compiled through machine learning techniques, and may include at least context information and user history information, as well as other information suitable for machine learning. A suggested match may include the input entity name and at least one identified entity name. In some embodiments, the at least one identified entity name may include a list of identified entity names. Thus, smart matching engine may identify multiple suggested matches between an input entity names. A suggested match may further include additional entity match information.

[064] Additional entity match information may include context information related to the natural language request, the input entity name, and the identified entity name. For example, for a user that lives in New York, an entity matching model may associate the entity name "taxi" in the request "call a taxi" with a particular taxi company in New York. When creating a suggested match for the input entity name "taxi," for that user, smart matching engine 270 may include the additional entity match information of the user's location as part of the suggested match. This may permit the system to disambiguate the term "taxi" differently based on the user's location.

[065] Context information usable by smart matching engine 270 may include a request context. For example, if a user makes a request to "call James" during the business day, smart matching engine 270 may suggest a match between the input entity name James and a business contact named James. If the request is made after hours, smart matching engine 270 may suggest a match with a contact from a friends and family contact list. Similarly, when attempting to disambiguate requests for directions, smart matching engine may suggest matches between locations that are closer to a present location of the user.

[066] User history information usable by smart matching engine 270 may include information about previous user requests and activity. For example, when requesting to make a call, smart matching engine 270 may select a name from a user's contact list that the user calls very frequently. When seeking directions, smart matching engine 270 may suggest a destination location that matches a location the user has travelled to frequently.

[067] Smart matching engine 270 may additionally employ machine learned matching techniques. For example, a user may frequently refer to a contact name by a nickname, making a request to "call Jim," where the name "James" is stored in the user's contact list. Such a connection may become a part of an entity matching model based on machine learning techniques, as described in more detail below.

[068] In some implementations smart matching engine 270 may suggest a match between an input entity name and an identified entity name based on an exact match. For example, where a user requests to "call John," the system may suggest only matches that match "John" exactly. In some implementations smart matching engine 270 may suggest a match between an input entity name and an identified entity name based on a phonetic match. For example, where a user requests to "call John," the system may suggest all matches that sound like "John," including, for example, "Jon." In some implementations smart matching engine 270 may suggest a match between an input entity name and an identified entity name based on similar sounds. Continuing to use the above example, smart matching engine 270 may suggest a match between "John" and "Juwana." In some implementations smart matching engine 270 may suggest a match between an input entity name and an identified entity name based on an interpretation of a user's pronunciation style.

[069] In some implementations smart matching engine 270 may suggest a match between an input entity name and an identified entity name based on learned associations. For example, a user may request to "call my brother." Smart matching engine 270 may suggest a match between the input entity name "my brother," and all user contacts having a same last name as the user. Smart matching engine 270 may also request clarification from the user about the entity name "my brother." Once the system has successfully confirmed a match between a user contact and the input entity name "my brother," the entity matching model may be updated with the information in order to correctly link the input entity name "my brother" to the appropriate contact. In another example, a user may request directions to "the pizza place." Smart matching engine may propose a suggested match between the input entity name "the pizza place," and the closest pizza restaurant. Once the system has successfully confirmed a match between "the pizza place," and a particular pizza restaurant, the entity matching model may be updated to associate "the pizza place," with the specific pizza restaurant.

[070] In some implementations, smart matching engine 270 may pass a suggested match between an input named entity and an identified named entity to confirmation engine 271 for command confirmation, as described below.

[071] In some implementations, smart matching engine 270 may determine an alternate identified entity name if confirmation engine 271 is unable to confirm the suggested match between the input named entity and the identified entity name. In some implementations, smart matching engine 270 may determine an alternate identified entity name using similar methods to those described above for selecting the first identified entity name. In determining an

alternate identified entity name, smart matching engine 270 may select a next best match for an input entity name. In some implementations, smart matching engine 270 may develop an n-best list of suggested matches for an input entity name including n number of potential matches ranked in order of best fit. Smart matching engine 270 may provide the 2nd, 3rd, 4th, etc., matches from an n-best list, in order, to confirmation engine 271 as alternate identified names if the first suggestion is cancelled or met with non-confirmation. In some implementations, smart matching engine 270 may provide an entire n-best list to confirmation engine 271.

[072] Confirmation engine 271 may include additional instructions that program computer system 110. The instructions of confirmation engine 271 may include, without limitation, instructions for confirming a suggest match between a received natural language input and a named entity, and/or other instructions that computer system 110 to perform various operations, each of which are described in greater detail herein.

[073] Confirmation engine 271 may perform command confirmation based on a user's confirmation history of a proposed entity match. Command confirmation may be performed actively and/or passively. For example, active command confirmation may include querying the user to confirm a suggest match - e.g., where a user has asked to "call James," a command confirmation may query the user "did you mean James Smith?" In some implementations, passive command confirmation may include permitting a user a predefined amount of time to cancel an action. For example, where a user has asked to "call James," a command confirmation may alert the user "calling James Smith," and wait for a predetermined amount of time to give the user a chance to cancel the action before executing the action. If the user does not cancel the action, then the suggested entity match may be confirmed. A predetermined amount of time may be relatively short, e.g., 1 to 5 seconds, or relatively long, e.g., 5-15 seconds. Selecting which command confirmation method to use may be based on a user's confirmation history.

[074] For example, confirmation engine 271 may use a history of a user's confirmation between a particular input entity name and a particular identified entity name. For example, a user may ask to "call James," and suggestion engine 270 may determine a suggested match between the input entity name "James" and an identified entity name "James Smith," corresponding to a user contact. If suggestion engine 270 has previously suggested the same match, and the user has previously provided confirmations, confirmation engine 271 may alert the user of the impending action. If the user has not previously provided enough confirmations of the suggested match, confirmation engine 271 may request active confirmation from the user.

Confirmation engine 271 may switch between these two methods when a number of previous confirmations of the suggested match exceeds a predetermined threshold number of times. Such a predetermined threshold number may be one or more times. A predetermined threshold number may vary depending on the type of entity name that is being matched (e.g., whether it is a contact name or a location address.)

[075] In some implementations, confirmation engine 271 may update a number of previous confirmations of a suggested match in a user's confirmation history after confirmation or non-confirmation is received from the user. If confirmation is received from the user for a particular suggested match, confirmation engine 271 may increment the number of previous confirmations. If a user cancels or fails to confirm a suggested match, confirmation engine 271 may instead decrement the number of previous confirmations. In some implementations, confirmation engine 271 may reset a number of previous confirmations to zero after a non-confirmation is received. In some implementations, confirmation engine 271 may separately record a number of non-confirmations and decrement or reset the previous confirmation score after a threshold number of non-confirmations is reached. In some implementations, confirmation engine 271 may separately record a number of consecutive non-confirmations and decrement or reset the previous confirmation score after a threshold number of consecutive non-confirmations is reached. In some implementations, confirmation engine 271 may do nothing to a previous confirmation score when a non-confirmation is received.

[076] In some implementations, confirmation engine 271 may request an alternate identified match from smart matching engine 270 if a non-confirmation or cancellation is received. Smart matching engine 270 may provide a next best alternate identified match to confirmation engine 271. Confirmation engine 271 may treat the alternate identified match similarly to the initial identified match, and determine an active or passive confirmation method based on a number of previous confirmations.

[077] In some implementations, confirmation engine 271 may receive an n-best list from smart matching engine 270. Confirmation engine 271 may then try to confirm with the user each entry in the n-best list as a selected match. In some implementations, confirmation engine 271 may try each member of the n-best list in turn, using the above-discussed confirmation methods based on confirmation history. In some implementations, confirmation engine 271 may try to confirm the first entry of an n-best list, and, if confirmation fails, query the user to select from the remaining entries in the n-best list.

[078] Learning engine 272 may include additional instructions that program computer system

110. The instructions of learning engine 224 may include, without limitation, instructions for providing machine learning updates to a smart matching model used by smart matching engine 270 and confirmation engine 271 in suggesting and confirming matches between natural language inputs and named entities, and/or other instructions that computer system 110 to perform various operations, each of which are described in greater detail herein.

[079] Learning engine 272 may include instructions for updating an entity matching model based on results provided by smart matching engine 270 and confirmation engine 271. As described above, smart matching engine 270 may suggest a match between an input entity name and an identified entity name, while confirmation engine 271 may operate to confirm the identified entity name and determine it as a selected entity name. When an entity name is confirmed as a selected entity name, learning engine 272 may update the entity matching model with information about the confirmed match. Learning engine 272 may update context information of the entity matching model with information about the context (e.g., time, location, whether the user is traveling, etc.) under which the suggested match was confirmed. Learning engine 272 may update user history information of the entity matching model, for example, to keep track of a frequency of user requests involving the selected entity name.

[080] For example, where a user has a requested to make a phone call or send a text message to a contact and confirmation engine 271 confirms the suggested match provided by the smart matching engine 270, learning engine 272 may update the entity matching model with information about the match between the input entity name and the identified entity name, including, for example, time of day, location of the user, category of the entity name (e.g., business or personal contact), and any other suitable information. In some implementations, when confirmation engine 271 obtains a non-confirm result of a suggested match, learning engine 272 may update entity matching model with information that the match was not confirmed, as well as other context information associated with the failed match.

[081] For example, a user may frequently his brother "James Lewis" in the evening but call his business associate "James Smith" during the day. Due to a frequency of calls, smart matching engine 270 may initially suggest a match between the request to "call James," and "James Smith," even when the call is taking place after 9 pm. After one or more non-confirmations between a command to "call James" at 9 pm and "James Smith," learning engine 272 may sufficiently update the entity matching model to make an association between "James" and the user's brother "James Lewis" in the evening, and between "James" and the user's business associate "James Smith" during the day.

[082] Execution engine 273 may include additional instructions that program computer system 110. The instructions of execution engine 273 may include, without limitation, instructions for executing a natural language request, and/or other instructions that computer system 110 to perform various operations, each of which are described in greater detail herein.

[083] Execution engine 273 may be configured to cause a computer system on which it is instantiated to carry out the action indicated by the request and the selected entity name determined from the processed natural language input . Execution engine 273 may be configured to engage any systems of the device or computer system on which it is instantiated to carry out the natural language request.

[084] Furthermore, as previously noted, some or all aspects of the foregoing operations may be performed by a given component illustrated in FIG. 1. In particular, some or all aspects of the foregoing operations may be performed by computer system 110 and/or user device 140. In some instances, some operations relating to natural language processing may be performed by user device 140, while other operations are performed at computer system 110.

[085] In some instances, the same operation relating to natural language processing may be performed by both user device 140 and computer system 110 (i.e., both user device 140 and computer system 110 may perform one or more of the operations relating to natural language processing). In implementations where NLP system 106 operates in a hybrid manner (e.g., one or more components of NLP system 106 are included in and operate at different remote devices such as computer system 110 and user device 140), communication manager 202 may mediate communication through a network, such as network 102.

[086] For example, a version of ASR engine 210 operating and executing on user device 140 may convert spoken utterances into text, which is then provided to a version of IRE 220 operating and executing on computer system 110. Alternatively or additionally, a version of ASR engine 210 operating and executing on user device 140 may convert spoken utterances into text and a version of ASR engine 210 operating and executing on computer system 110 may also convert the spoken utterances into text, in which case one result may be chosen over.

[087] A more detailed description of such hybrid natural language processing is disclosed in U.S. Patent Application Serial No. 12/703,032, entitled "System and Method for Hybrid Processing in a Natural Language Voice Services Environment," filed on February 9, 2010, the disclosure of which is incorporated in its entirety herein. The foregoing or other hybrid systems may be used for NLP system 106.

[088] FIG. 3 depicts a process 300 of natural language processing disambiguation, according

to an implementation of the invention. The various processing operations and/or data flows depicted in FIG. 3 (and in the other drawing figures) are described in greater detail herein. The described operations may be accomplished using some or all of the system components described in detail above and, in some implementations, various operations may be performed in different sequences and various operations may be omitted. Additional operations may be performed along with some or all of the operations shown in the depicted flow diagrams. One or more operations may be performed simultaneously. Accordingly, the operations as illustrated (and described in greater detail below) are exemplary by nature and, as such, should not be viewed as limiting.

[089] In an operation 302, process 300 may include a natural language input operation. In natural language input operation 300 NLP system 106 receives a natural language input, such as a verbal utterance, from the user. The utterance may be received, for example, via a microphone incorporated into user device 140. ASR manager 210 of NLP subsystem 106 may receive the utterance.

[090] In an operation 304, process 300 may include an natural language processing operation. Operation 304 may perform natural language processing on the received natural language input. Components of NLP subsystem 106 may cooperate to process the received natural language input to determine at least a request and an entity name. For example, as described above with respect to Fig. 2, ASR engine 210 may perform automatic speech recognition on the natural language input and IRE 220 may perform intent recognition on the natural language input. NLP subsystem 106 may thus determine a request and an input entity name from the user's natural language input.

[091] In an operation 306, process 300 may include a smart matching step. During a smart matching operation 306, smart matching engine 270 may determine a suggested match between the input entity name and an identified entity name. Smart matching engine 270 may access an entity matching model, including context information, user history, and machine learned user tendencies to determine a suggested match.

[092] In an operation 308, process 300 may include an entity match confirmation step. An entity match confirmation operation may be performed by confirmation engine 271. Confirmation engine 271 may determine whether to designate the identified entity name as a selected entity name, based on a user's confirmation history of suggested matches between the identified entity name and the input entity name. The user's confirmation history may be used to determine a method, e.g., active or passive, by which confirmation engine 271 confirms the

entity name. Entity match confirmation operation 308 may be understood in more detail with reference to Fig. 4.

[093] FIG. 4 depicts a process 400 of natural language match confirmation, according to an implementation of the invention. The various processing operations and/or data flows depicted in FIG. X (and in the other drawing figures) are described in greater detail herein. The described operations may be accomplished using some or all of the system components described in detail above and, in some implementations, various operations may be performed in different sequences and various operations may be omitted. Additional operations may be performed along with some or all of the operations shown in the depicted flow diagrams. One or more operations may be performed simultaneously. Accordingly, the operations as illustrated (and described in greater detail below) are exemplary by nature and, as such, should not be viewed as limiting.

[094] In an operation 401, process 400 may include a suggested match reception step. During suggested match reception operation, confirmation engine 271 may receive information about a suggested match from smart matching operation 270. Confirmation engine 271 may receive the input entity name and the suggested match for it, i.e., the identified entity name.

[095] In an operation 402, process 400 may include a match confirmation history determination step. Match confirmation history determination operation 402 may access a user history database 180/190 to determine a user command confirmation history associated with the suggested match. The user command confirmation history may include a previous confirmations counter comprising information about a user's previous confirmations of the suggested match between the input entity name and identified entity name. For example, a user's confirmation history may include the information that a user has confirmed a particular match 9 times, cancelled that match 2 times, and confirmed the previous 6 suggestions of the match.

[096] In some implementations, the user command confirmation history may further include additional entity match information, which may include the context of a user's confirmation history with respect to a suggested match. For example, additional entity match information may include information about circumstances under which a user has confirmed a suggested match. For example, where a user has made the request to "call James," and the suggested match is with the identified entity "James Smith" the business associate, the additional entity match information may include the information that the user has confirmed this suggested match 13 times during business hours, but cancelled the match 7 times during evening hours.

During match confirmation history determination operation 402, confirmation engine 271 may determine the number of previous confirmations by a user of the suggested match between the input entity name and the identified entity name, and may further use additional entity match information to make the determination.

[097] In an operation 404, process 400 may include a confirmation history thresholding step. Confirmation history thresholding operation 404 may include comparing the number of previous confirmations by the user to a predetermined threshold number of confirmations. Confirmation history thresholding operation 404 may include determining whether the previous number of confirmations exceeds or fails to exceed the predetermined threshold number of confirmations.

[098] In an operation 406, process 400 may include an active match confirmation step. If confirmation engine 271 determines that a number of previous confirmations does not exceed a predetermined confirmation threshold number, process 400 may proceed to an active confirmation request operation 406. During active confirmation request operation 406, confirmation engine 271 may cause the host system to query the user for active confirmation of the suggested match provided by smart matching engine 270. For example, where a user has requested directions to the input entity name "233 Main St.," the system may actively request that the user confirm the identified entity name by querying the user "Did you mean 233 Main St, Cambridge, Massachusetts?" When a match has been confirmed by the user, confirmation engine 271 may designate the confirmed entity name as a selected entity name for subsequent execution. Confirmation engine 271 may further update the previous confirmation history with the information that the suggested match has been confirmed. Such updating may include incrementing a number of previous confirmations, incrementing a number of previous consecutive confirmations, and/or providing any additional entity match information about the context in which the suggested match was confirmed.

[099] When a match has been cancelled or declined by the user, confirmation engine 271 may designate the identified entity name as a non-match, and proceed to an alternative match suggestion operation 410. Confirmation engine 271 may further update the previous confirmation history with the information that the suggested match was not confirmed. Such updating may include decrementing a number of previous confirmations, resetting a number of previous consecutive confirmations, resetting a total number of previous confirmations, and/or providing any additional entity match information about the context in which the suggested match was cancelled.

[01 00] In an operation 410, process 400 may include an alternative match suggestion step. Confirmation engine 271 may perform an alternative match suggestion operation after a user has cancelled or failed to confirm an initially suggested match provided by smart matching engine 270 (either in operation 406 or an operation 412). In some implementations, confirmation engine 271 may request that smart matching engine 270 provide a next best match for the input entity name. Confirmation engine 271 may then suggest this next best match as an alternate identified entity name to the user for confirmation. In some implementations, confirmation engine 271 may select a next best match from an n-best list that has already been provided by smart matching engine 270. Confirmation engine 271 may then suggest this next best match as an alternate identified entity name to the user for confirmation. In some implementations, confirmation engine 271 may provide, to the user, multiple potential matches, e.g., from an n-best list provided by smart matching engine 270, from which the user may select an alternate identified entity name for confirmation. After alternate match suggestion step 410, process flow may return to catch confirmation history determination operation 402 and repeat steps 402, 404, 406, and/or 412 using the alternate identified entity name.

[01 01] In an operation 412, process 400 may include a passive confirmation operation. If confirmation engine 271, at thresholding step 404, determines that a number of previous confirmations exceeds a predetermined confirmation threshold number, process 400 may proceed to a passive confirmation request operation 412. A passive confirmation operation 412 may be performed by confirmation engine 271 when the user has previously confirmed the suggested match more than a certain number of times. After multiple past confirmations, the system may determine a higher level of certainty in the suggested match, and not require that the user actively confirm the match. A passive confirmation operation may proceed by alerting the user (e.g., by an audio prompt or a display prompt) that an action based on the request and the identified entity name will be carried out. A predetermined pause period may then be introduced to give the opportunity to the user to cancel the action. If the user does not cancel the action, then confirmation engine 271 may confirm the suggested match. If the user does issue a cancellation command, then confirmation engine 271 may designate the suggested match as non-confirmed.

[01 02] When a match has been confirmed by the user, confirmation engine 271 may designate the confirmed entity name as a selected entity name for subsequent execution at operation 414. Confirmation engine 271 may further update the previous confirmation history with the information that the suggested match has been confirmed. Such updating may include

incrementing a number of previous confirmations, incrementing a number of previous consecutive confirmations, and/or providing any additional entity match information about the context in which the suggested match was confirmed.

[01 03] When a match has been cancelled or non-confirmed by the user, confirmation engine 271 may designate the identified entity name as a non-match, and proceed to an alternative match suggestion operation 410. Confirmation engine 271 may further update the previous confirmation history with the information that the suggested match was not confirmed. Such updating may include decrementing a number of previous confirmations, resetting a number of previous consecutive confirmations, resetting a total number of previous confirmations, and/or providing any additional entity match information about the context in which the suggested match was cancelled.

[01 04] In an operation 414, process 400 may include a selected match finalization operation. Selected match finalization operation 414 may be performed by confirmation engine 271 to pass the confirmed selected entity name to execution engine 273. During operation 414, information about the confirmed match, e.g., the selected entity name and any additional match entity information, may be passed to learning engine 272 to update the entity matching model.

[0105] . Returning now to Fig. 3, in an operation 310, process 310 may include a natural language execution step. After confirmation of the identified entity name as the selected entity name during command confirmation operation 308, natural language execution operation 310 may be performed by execution engine 273 to cause the system to carry out the request on the selected entity name. For example, the request may include a phone dialing request and the selected entity name may be a user object. Execution of this natural language result would cause a device of the user to initiate a phone call. The request may include a search request, and the selected entity name may include a search object. Execution of this natural language result would cause a device of the user to carry out a search for information about the search object. The request may include a navigation request, and the selected entity name may include a street address or other physical location. Execution of this statement would cause the device of the user to provide the user with directions to the street address or physical location.

[01 06] Other implementations, uses and advantages of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The specification should be considered exemplary only, and the scope of the invention is accordingly intended to be limited only by the following claims.

CLAIMS

What is claimed is:

1. A computer implemented method of natural language disambiguation, the method being implemented in a computer system having one or more physical processors programmed with computer program instructions that, when executed by the one or more physical processors, cause the computer system to perform the method, the method comprising:

processing, by the computer system, using an intent recognition engine, a natural language input;

determining, by the computer system, based on the processing, at least a type of request and an input entity name;

identifying, by the computer system, at least a first match between the input entity name and a first identified entity name and at least a second match between the input entity name and a second identified entity name;

requesting, by the computer system, a confirmation of whether the first identified entity name or the second identified entity name was intended by the natural language input;

receiving, by the computer system, a response to the requested confirmation;

generating, by the computer system, a request based on the type of request and the response; and

storing, by the computer system, information indicating the response for use in future natural language input processing in a command confirmation history.

2. The computer implemented method of claim 1, wherein the input entity name includes at least one of a contact name, an address, or a business name.

3. The computer implemented method of claim 1, wherein the type of request includes a phone dialing request and the first identified entity name or the second identified entity name includes a user contact.

4. The computer implemented method of claim 1, wherein the type of request includes a search query and the first identified entity name or the second identified entity name includes a search object.

5. The computer implemented method of claim 1, wherein the type of request includes a navigation query and the first identified entity name or the second identified entity name includes a street address.
6. The computer implemented method of claim 1, wherein identifying at least a first match between the input entity name and the first identified entity name and at least a second match between the input entity name and the second identified entity name is at least partially based on at least one of a user location, a time of day, a user request history, a user request frequency, and a user preference.
7. The computer implemented method of claim 1, wherein identifying at least a first match between the input entity name and the first identified entity name and at least a second match between the input entity name and the second identified entity name is at least partially based on a user preference for contacting at least one of family, friends, and business associates.
8. The computer implemented method of claim 1, wherein identifying at least a first match between the input entity name and the first identified entity name and at least a second match between the input entity name and the second identified entity name is at least partially based on the command confirmation history indicating a number of previous responses that indicate whether the first identified entity name or the second identified entity name was intended by the natural language input.
9. The computer implemented method of claim 8, wherein requesting a confirmation of whether the first request or the second request was intended by the natural language input when the command confirmation history indicates a number of previous responses that indicate whether the first request or the second request was intended by the natural language input above a predetermined threshold includes:
 - selecting the first request or the second request as an intended request based on the command confirmation history,
 - providing information about the intended request as the response to the requested confirmation; and

the method further comprises:
alerting the user that execution of the intended request will be performed;
waiting, by the computer system, a predetermined amount of time for user cancellation of the announced execution prior to executing the intended request; and
updating the number of previous responses received of whether the first request or the second request was intended by the natural language input stored in the command confirmation history

10. The computer implemented method of claim 9, further comprising:

receiving, by the computer system, a cancellation request from the user;
requesting that the user select an alternate request different from the intended request ;
and
storing the alternate request selected by the user for use in future natural language input processing in a command confirmation history.

11. The computer implemented method of claim 1, wherein storing information indicating the response for use in future natural language input processing in a command confirmation history comprises:

updating, based on the response, a first number of times that the first identified entity name was intended or a second number of times that the second identified entity name was intended; and

wherein the method further comprises:

processing, by the computer system, using the intent recognition engine, a second natural language input;

determining, by the computer system, a second input entity name based on the processing;

identifying, by the computer system, at least a third match between the second input entity name and the first identified entity name and at least a fourth match between the second input entity name and the second identified entity name;

determining, by the computer system, that the first number of times that the first identified entity name was intended exceeds a predetermined threshold;

determining, by the computer system, that the first identified entity name is intended by the second natural language input based on the determination that the first number of times that the first identified entity name was intended exceeds the predetermined threshold.

12. The computer implemented method of claim 1, wherein requesting a confirmation comprises:

requesting that the user provide an alternate entity name, different from the first identified entity name and the second identified entity name, ;

providing information about the alternate entity name as the response to the requested confirmation, and wherein

the method further comprises

storing, by the computer system, information indicating the response for use in future natural language input processing in a command confirmation history .

13. The computer implemented method of claim 1, wherein storing information indicating the response for use in future natural language input processing further includes updating the intent recognition engine with information indicating a match between the input entity name and the first identified entity name or second identified entity name associated with the confirmed first request or second request,

wherein the computer system recognizes the first identified entity name and the second identified entity name as being an exact match.

14. The computer implemented method of claim 1, wherein storing information indicating the response for use in future natural language input processing further includes updating the intent recognition engine with information indicating a match between the input entity name and the first identified entity name or second identified entity name associated with the confirmed first request or second request,

wherein the computer system recognizes the first identified entity name and the second identified entity name as being a phonetic match.

15. The computer implemented method of claim 1, wherein storing information indicating the response for use in future natural language input processing further includes updating the intent

recognition engine with information indicating a match between the input entity name and the first identified entity name or second identified entity name associated with the confirmed first request or second request,

wherein the computer system recognizes the first identified entity name and the second identified entity name as being a partial match.

16. A system of natural language disambiguation, the system comprising:
 - a computer system comprising one or more physical processors programmed by computer program instructions that, when executed, cause the computer system to:
 - process, using an intent recognition engine, a natural language input;
 - determine, based on the processed natural language input, at least a type of request and an input entity name;
 - identifying at least a first match between the input entity name and a first identified entity name and at least a second match between the input entity name and a second identified entity name;
 - request a confirmation of whether the first identified entity name or the second identified entity name was intended by the natural language input;
 - receive a response to the requested confirmation;
 - generate a request based on the type of request and the response; and
 - store information indicating the response for use in future natural language input processing in a command confirmation history.

17. The system of claim 16, wherein the input entity name includes at least one of a contact name, an address, and a business name.

18. The system of claim 16, wherein the request includes a phone dialing request and the and the first identified entity name or the second identified entity name includes a user contact.

19. The system of claim 16, wherein the type of request includes a search query and the and the first identified entity name or the second identified entity name includes a search object.

20. The system of claim 16, wherein the type of request includes a navigation query and the first identified entity name or the second identified entity name includes a street address.

21. The system of claim 15, wherein the computer program instructions further cause the computer system to:

identify at least the first match between the input entity name and the first identified entity name and at least the second match between the input entity name and the second identified entity name at least partially based on at least one of a user location, a time of day, a user request history, a user request frequency, and a user preference.

22. The system of claim 15, wherein the computer program instructions further cause the computer system to:

identify at least the first match between the input entity name and the first identified entity name and at least the second match between the input entity name and the second identified entity name at least partially based on user preferences including a preference for contacting at least one of family, friends, and business associates.

23. The system of claim 15, wherein the computer program instructions further cause the computer system to:

identify at least the first match between the input entity name and the first identified entity name and at least the second match between the input entity name and the second identified entity name at least partially based on the command confirmation history indicating a number of previous responses that indicate whether the first identified entity name or the second identified entity name was intended by the natural language input.

24. The system of claim 15, wherein the instructions to request a confirmation of whether the first request or the second request was intended by the natural language input when the when the command confirmation history indicates a number of previous responses above a predetermined threshold further include instructions to cause the at least one processor to:

select the first request or the second request as an intended request based on the command confirmation history,

provide information about the intended request as the response to the requested confirmation; and

the computer program instructions further cause the computer system to:

alert the user that execution of the intended request will be performed;

wait a predetermined amount of time for user cancellation of the announced execution prior to executing the intended request; and

update the number of previous responses received of whether the first request or the second request was intended by the natural language input stored in the command confirmation history.

25. The system of claim 24, further comprising instructions to cause the computer system to:

receive a cancellation request from the user;

request that the user select an alternate request different from the intended request ; and

store the alternate request selected by the user for use in future natural language input processing in a command confirmation history.

26. The system of claim 15, wherein the instructions to store information indicating the response of use in future natural language processing in a command confirmation history further cause the computer system to:

update, based on the response, a first number of times that the first identified entity name was intended or a second number of times that the second identified entity name was intended;

and wherein the computer program instructions further cause the computer system to:

process, using the intent recognition engine, a second natural language input;

identify a second input entity name based on the processing;

identify at least a third match between the second input entity name and the first identified entity name and at least a fourth match between the second input entity name and the second identified entity name;

determine that the first number of times that the first identified entity name was intended exceeds a predetermined threshold;

determine that the first identified entity name is intended by the second natural language input based on the determination that the first number of times that the first identified entity name was intended exceeds the predetermined threshold .

27. The system of claim 15, wherein the instructions to request a confirmation of further cause the computer system to:

request that the user provide an alternate entity name, different from the first identified entity name and the second identified entity name, ;

provide information about the alternate entity name as the response to the requested confirmation, and wherein

the computer system is further caused to store information indicating the response for use in future natural language input processing in a command confirmation history .

28. The system of claim 15, wherein the instructions to store information indicating the response of use in future natural language processing in a command confirmation history further cause the computer system to:

update the intent recognition engine with information indicating a match between the input entity name and the first identified entity name or second identified entity name associated with the confirmed first request or second request,

wherein the computer system recognizes the first identified entity name and the second identified entity name as being an exact match.

29. The system of claim 15, wherein the instructions to store information indicating the response of use in future natural language processing in a command confirmation history further cause the computer system to:

update the intent recognition engine with information indicating a match between the input entity name and the first identified entity name or second identified entity name associated with the confirmed first request or second request,

wherein the computer system recognizes the first identified entity name and the second identified entity name as being a phonetic match.

30. The system of claim 15, wherein the instructions to store information indicating the response of use in future natural language processing in a command confirmation history further cause the computer system to:

update the intent recognition engine with information indicating a match between the input entity name and the first identified entity name or second identified entity name associated with the confirmed first request or second request,

wherein the computer system recognizes the first identified entity name and the second identified entity name as being a partial match.

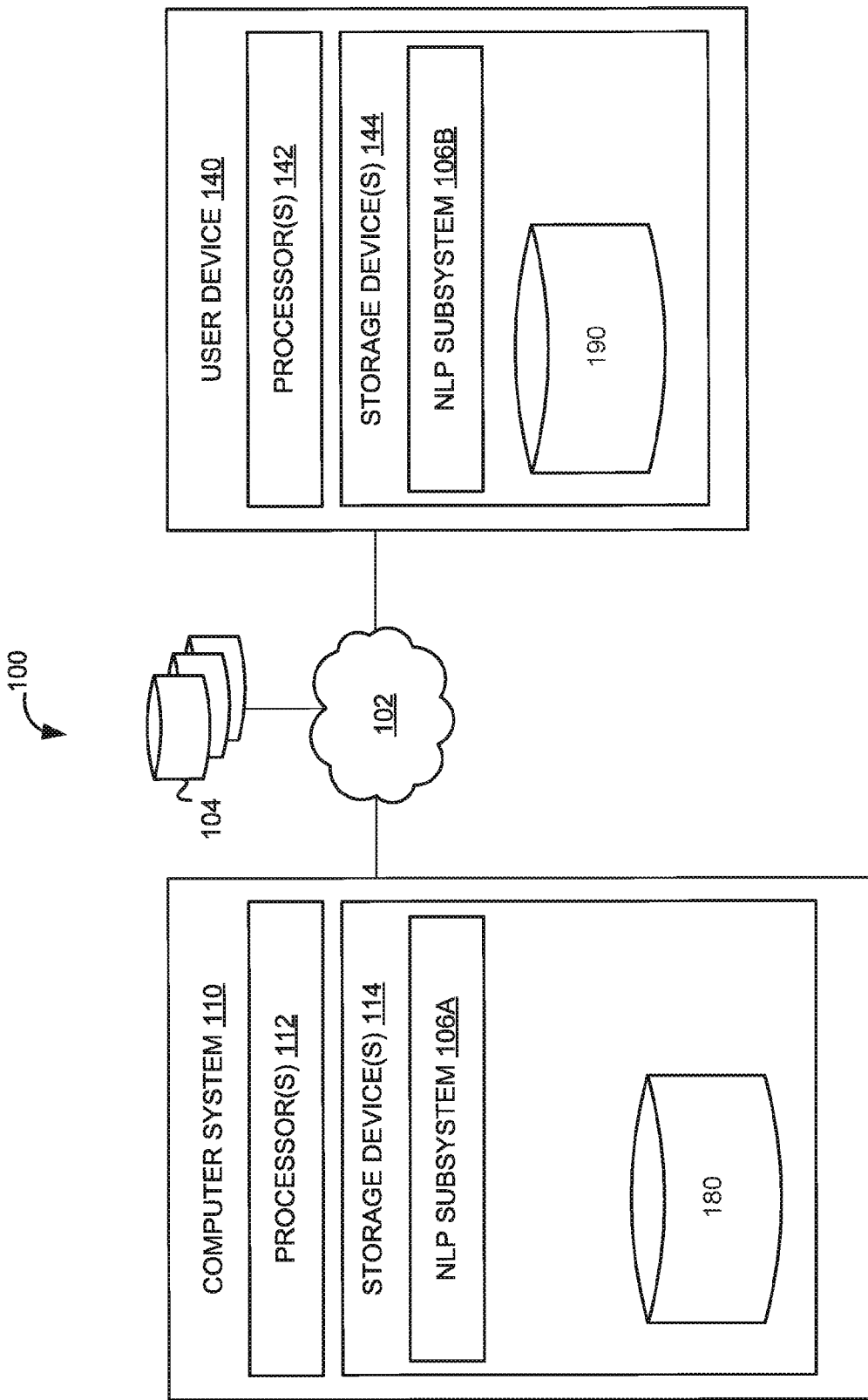


FIG. 1

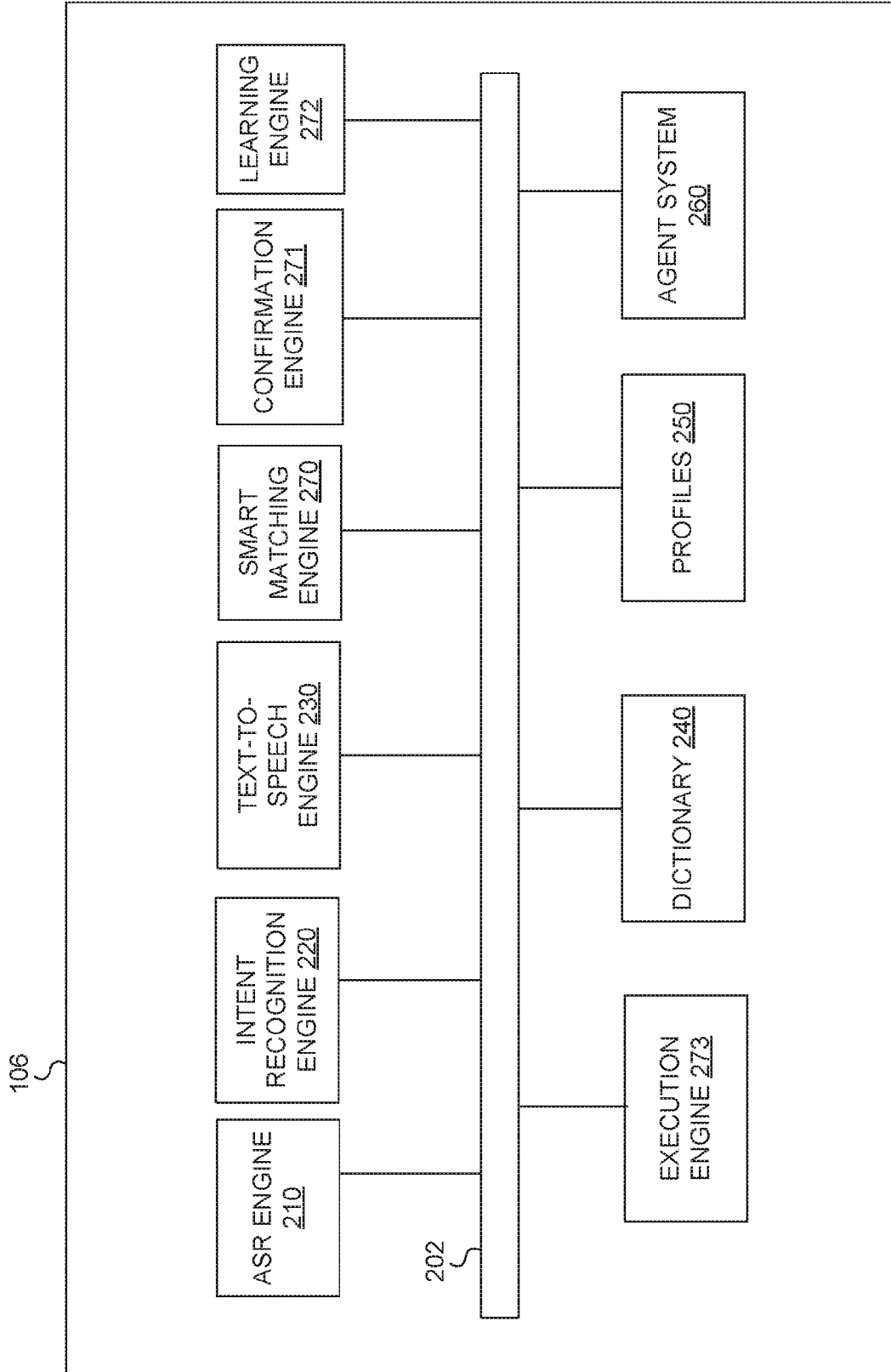


FIG. 2

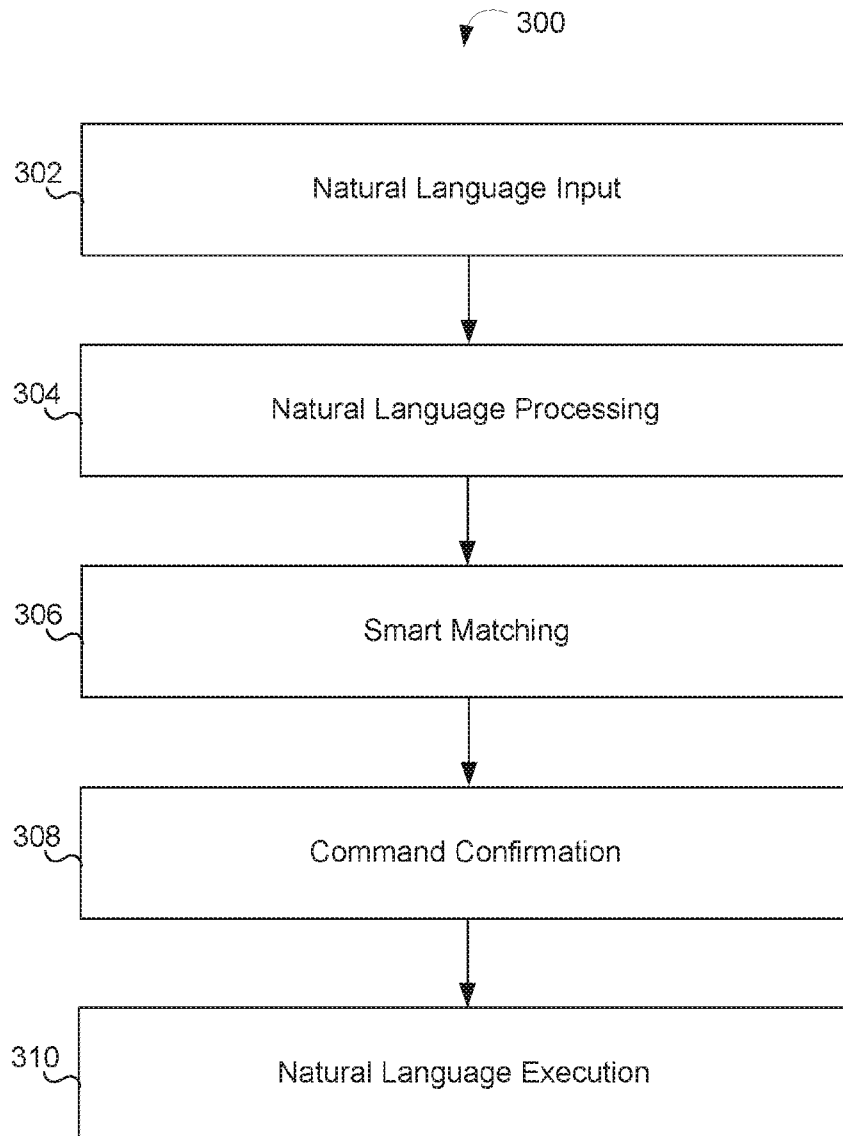


FIG. 3

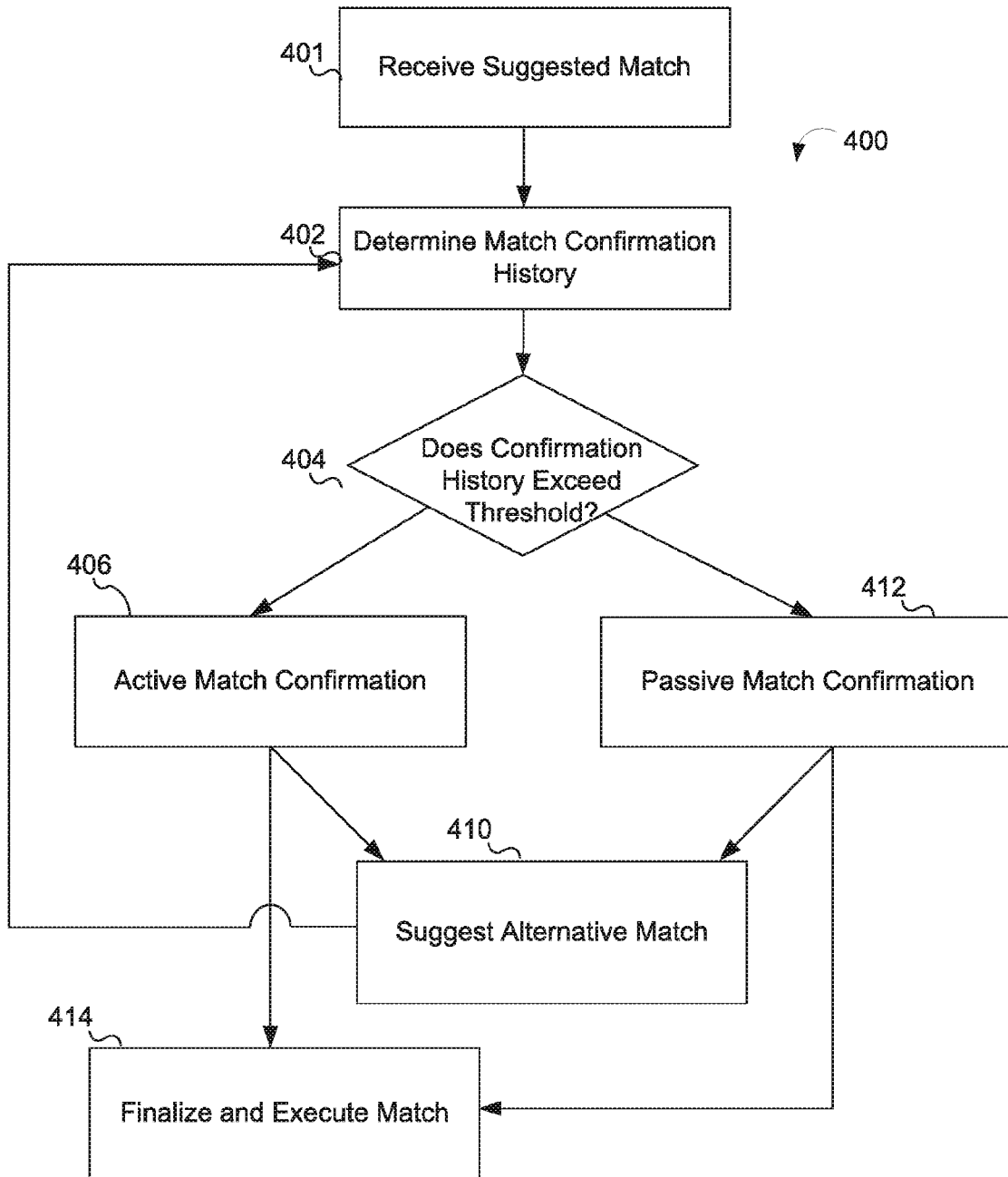


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 17/44610

<p>A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 17/27 (201 7.01) CPC - G06F 17/2785, G06F 17/274, G06F 17/271, G06F 17/277, G06F 17/2755, G10L 15/22, G 10L 15/1 83, G10L 15/1 822, G 10L 15/1 9, G 10L 15/1 93, G06F 17/30654, G06F 17/30684, G06F 17/2785, G06F 17/30672, G06F 17/3064, G06F 17/30663, G06F 17/30616, G06F 17/30672</p>		
<p>According to International Patent Classification (IPC) or to both national classification and IPC</p>		
<p>B. FIELDS SEARCHED</p>		
<p>Minimum documentation searched (classification system followed by classification symbols) See Search History Document</p>		
<p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched See Search History Document</p>		
<p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) See Search History Document</p>		
<p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X -- Y	US 2015/0066479 A1 (Maluuba Inc.) 02 March 2015 (02.03.2015), entire document especially abstract; para [0004], [0007], [0073], [0074], [0081]-[0083], [01 19]-[0152], [0222]-[0252], [0263]	1-12, 14-27, 29, 30 ----- 13, 28
Y	US 2009/0171912 A1 (Nash et al.) 02 July 2009 (02.07.2009), para [0021], [0027], [0032], [0034], [0038]	13, 28
<p> Further documents are listed in the continuation of Box C. See patent family annex.</p>		
<p>* Special categories of cited documents:</p>		
"A"	document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E"	earlier application or patent but published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means	...&... document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed	
Date of the actual completion of the international search	Date of mailing of the international search report	
14 September 2017 (14.09.2017)	17 OCT 2017	
Name and mailing address of the ISA/US	Authorized officer:	
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774	