

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

(19) World Intellectual Property Organization

International Bureau



(10) International Publication Number

WO 2013/134871 A1

(43) International Publication Date

19 September 2013 (19.09.2013)

WIPO | PCT

(51) International Patent Classification:

G06F 17/30 (2006.01) G06F 17/27 (2006.01)

(21) International Application Number:

PCT/CA2013/050181

(22) International Filing Date:

12 March 2013 (12.03.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/610,606 14 March 2012 (14.03.2012) US

(72) Inventors; and

(71) Applicants : FARMANER, Gary [CA/CA]; 633 Bay Street, Suite 1211, Toronto, Ontario M5G2G4 (CA). DICARLANTONIO, Ron [CA/JP]; #201, 5-8-11 Todoroki, Setagaya-ku, Tokyo (JP).

(74) Agent: BHOLE, Anil; 10 King Street East, Suite 1400, Toronto, Ontario M5C1C3 (CA).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

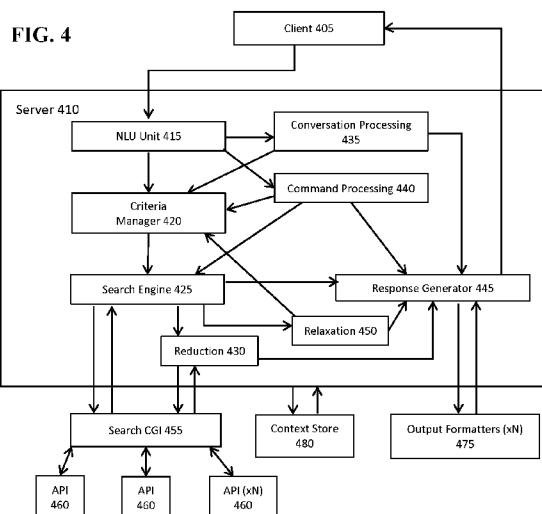
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

(54) Title: SYSTEM AND METHOD FOR CONVERSATION-BASED INFORMATION SEARCH



(57) **Abstract:** A system and method for performing an operation based on a contextual command, which operation further comprises interactively searching for information, comprising: receiving an input in a context, returning a result in respect of the received context by at least one of reducing, relaxing, and location handling in respect of the input value, and performing an operation based upon the context of the input criteria. Reducing comprises narrowing the total number of results by their contextual relevance, wherein the narrowing is comprised in dynamically generated real-time interactions. Relaxing further comprises, when an exact result is not found, broadening the input search criteria automatically, and where appropriate, obtaining a result. The location handling further comprises disambiguating addresses and locations where there are conflicts based on an input history, and establishing relationships within addresses based upon the input history.

SYSTEM AND METHOD FOR CONVERSATION-BASED INFORMATION SEARCH

CROSS REFERENCE

[0001] This application claims priority to United States patent application no. 61/610,606 filed March 14, 2012, which is incorporated herein by reference.

TECHICAL FIELD

[0002] The following relates generally to computerized searching and more specifically to contextual search.

BACKGROUND

[0003] There are two common forms of searching for information today: keyword driven search or call flow driven search.

[0004] Keyword driven search allows the user to search a large amount of data by inputting a search phrase either as a list of keywords or in some cases a natural language sentence and obtaining a list of highly likely related information. Often, a user is challenged with having to pick the perfect search phrase to get the exact information they are looking for. Typically, a very large list of information is provided to the user in a returned result and the user must decide herself how to adjust the search query to reduce this list of information to get the results she is interested in. Very little assistance is typically provided to the user for reducing this list to their needs, other than, for example, the seldom and well known, "Did you mean Jaguar?"

[0005] Call flow driven search allows the user to search for information through a pre-defined list of options. An example of this would be an automated phone system where the user is presented with a list of options to choose from and wherein the user cannot move forward without selecting an appropriate option from the list of presented options. Another example would be a website which allows users to select specific pre-defined categories to narrow their search results. This method provides an interactive method for finding information, which is easy to use, but typically limits the user to provide one piece of information at a time and follow a specific pre-designed flow of questions regardless of their needs. A significant time may be needed to develop and maintain an effective call flow that is both easy to use for the user and covers the data being searched sufficiently.

[0006] Since the data itself is changing based on the user location, their needs, and the databases being searched, a pre-designed call flow typically does not provide an efficient method (least amount of steps) to reach the desired results. Further, call flow driven searches are passive from the user's perspective, as the user is asked to follow directions in order to obtain relevant information.

[0007] Certain interactive, intelligent chat systems have been developed to address the foregoing, but these typically have been "authored" such that questions and scenarios were written specifically for each type (or domain) of content and context. This has typically required significant development time, effort and cost.

[0008] It is an object of the following to obviate or mitigate at least one of the above disadvantages.

SUMMARY

[0009] In one aspect, a computer automated system is provided, the system adapted to interpret commands contextually, and comprising a processing unit and memory element having instructions encoded thereon, which cause the system to: receive a user input command which corresponds to an instruction for performing an operation in a context; disambiguate the input command; perform the operation based on the disambiguated input command; and return in response to the disambiguated input command, zero or more results.

[0010] In another aspect, in a computer automated system adapted to interpret commands contextually, and comprising a processing unit and memory element having instructions encoded thereon, a method is provided, the method comprising: receiving a user input command which corresponds to an instruction for performing an operation in a context; disambiguating the input command; performing the operation based on the disambiguated input command; and returning in response to the disambiguated input command, zero or more results.

[0011] In yet another aspect, a system is provided, the system comprising a processing unit and memory element, and having instructions encoded thereon, which instructions cause the system to: receive an input in a context; return a result in respect of the received context by at least one of narrowing, broadening, and location handling in respect of the input value; wherein an operation is performed based upon the context of input criteria; wherein the narrowing further comprises returning a one or more relevant items based on a current result returned in response

to the input, and comprising possible criteria values; wherein the broadening further comprises, when an exact result is not found, broadening the input criteria automatically, and where appropriate, to obtain a result; and wherein the location handling further comprises disambiguating addresses and locations where there are conflicts based on an input history, and establishing relationships within addresses based upon an input history.

[0012] In a further aspect, a dynamic, self-evolving, computer automated system is provided, the system comprising a processing unit and a memory element, and having instructions encoded thereon which instructions cause the system to: develop evolving interactive capability without human authoring of scenarios for each user interaction, wherein said evolving further comprises determining a user interaction based on the context and content; automatically define rules that enhance the automated functionality; implement natural language processing wherein said natural language processing comprises mapping a user input to meaning, and which mapping further comprises genre tagging; differentiate between a set which comprises a grouping and a set which comprises the action of making a change; and create a hierarchical structure for allowing matching of input to at least one of a single or plurality of general ancestors and a single or plurality of descendants.

[0013] In yet a further aspect, in a computer automated system comprising a processing unit and a memory element, and having instructions encoded thereon, a method for dynamic self-evolving of the computer automated system is provided, the method comprising: developing evolving interactive capability without human authoring of scenarios for each user interaction, wherein said evolving further comprises determining a user interaction based on the context and content; automatically defining rules to enhance the automated functionality; implementing natural language processing wherein said natural language processing further comprises mapping a user input to meaning, and which mapping further comprises genre tagging; differentiating between a set which comprises a grouping and a set which comprises the action of making a change; creating a hierarchical structure for allowing matching to at least one of a single or plurality of general ancestors or a single or plurality of descendants.

BRIEF DESCRIPTION OF DRAWINGS

[0014] The features of the invention will become more apparent in the following detailed description in which reference is made to the appended drawings wherein:

[0015] FIG. 1 illustrates a process flow in an embodiment of a system that enables searching for content and information through conversational interaction.

[0016] FIG. 2 illustrates the process flow in an embodiment, of the reduction method.

[0017] FIG. 3 illustrates the process flow in an embodiment, of the relaxation method.

[0018] FIG. 4 illustrates components of a system in an embodiment.

DETAILED DESCRIPTION

[0019] Embodiments will now be described with reference to the figures. It will be appreciated that for simplicity and clarity of illustration, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements. In addition, numerous specific details are set forth in order to provide a thorough understanding of the embodiments described herein. However, it will be understood by those of ordinary skill in the art that the embodiments described herein may be practiced without these specific details. In other instances, well-known methods, procedures and components have not been described in detail so as not to obscure the embodiments described herein. Also, the description is not to be considered as limiting the scope of the embodiments described herein.

[0020] Reference in this specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments mutually exclusive of other embodiments. Moreover, various features are described which may be exhibited by some embodiments and not by others. Similarly, various requirements are described which may be requirements for some embodiments but not other embodiments.

[0021] It will also be appreciated that any module, unit, component, server, computer, terminal or device exemplified herein that executes instructions may include or otherwise have access to computer readable media such as storage media, computer storage media, or data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Computer storage media may include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information, such

as computer readable instructions, data structures, program modules, or other data. Examples of computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by an application, module, or both. Any such computer storage media may be part of the device or accessible or connectable thereto. Any application or module herein described may be implemented using computer readable/executable instructions that may be stored or otherwise held by such computer readable media.

[0022] This specification may be best understood with reference to the following terms, which are explained with the following non-limiting examples.

[0023] Natural Language – A human language, in contrast to a formal (i.e. specifically designed) language, such as a computer programming language. In the modern online world, natural language is affected by issues of spelling, grammar, colloquialisms, slang, abbreviations, emoticons, swearing, technical terms, acronyms, etc.

[0024] Natural Language Processing (NLP) – The conversion of a string in a natural language into a data structure, or formal language, that provides information about the string. This can include work tokenization, morphological analysis (e.g. parts of speech), and dialogue act (type of sentence), and general conversions of the input into a form more suitable for computation manipulation.

[0025] Natural Language Understanding (NLU) – A set of algorithms used to map an input in a natural language to a set of system state changes that reflect the effect the input is intended to cause.

[0026] Agent – A system capable of interaction using natural language, in an intelligent way, for a useful purpose.

[0027] Conversational Interaction – The set of inputs and outputs between the user and the Agent.

[0028] Small talk – Simple responses to user input meant to make the experience more enjoyable, and provide personality to the Agent.

[0029] Queries – Information requests on the current search candidate set that does not change the current search conditions (for example, “How far is this store from me?”).

[0030] Domain – The subject or subjects about which the Agent is prepared to interact.

[0031] Locale – The set of attributes related to the user’s current location. This can include position/location, default language, measurement units, date and time formats, etc.

[0032] Synonyms – Words or phrases which have a common meaning in the domain of operation.

[0033] Normal – A canonical name representing a set of Synonyms.

[0034] Family – A formal collection of related Normals.

[0035] Genre – A placeholder that represents a hierarchical family of related words. In one embodiment it consists of the combination of a Family and a Normal.

[0036] Genre Tagging or Input Genre – A representation of a word or sub-sentence of the input by a Genre with the attached word or sub-sentence.

[0037] Genrization – The process of Genre Tagging a string.

[0038] Genrized or Genre Tagged – Having had Genrization applied.

[0039] Genre Condition – A list of words and Genres that can be matched in any order.

[0040] Genre Grammar Condition – A sentence or sub-sentence consisting of words, Genres, and special meaning grammar tokens, and matched against genrized input for the performance of NLU.

[0041] Matching Genre – Any specific Genre in the Genre Condition or Genre Grammar Condition.

[0042] Genre Condition Match – The matching of the Genrized form of the user input with a Genre Condition or Genre Grammar Condition.

[0043] Key Genre – Those Genres of a Genre Condition that are used by the system to extract values for the target(s) of the NLU.

[0044] Associated Genre – Genres that if present may be considered to be also part of Genre Condition Match.

[0045] Criteria – A set of formally defined conditions represented via set of names, a sub-set of which can have one or more values applied for purposes of searching and/or controlling process flow. For ease of writing, Criteria can refer to the singular or plural.

[0046] Criteria Value – A single value for particular Criteria, which could be formalized (a canonical set of values) or “free input” wherein it takes on a value from user input or searched content (e.g. Store Name).

[0047] Collapsible or Drill-down Criteria – A Criteria whose Criteria Values are defined as a tree where the values become more specific the deeper in the value tree they appear. Collapsible Criteria are presented as lists flattened at a given tree depth, and can have the children presented (drill-down) to further restrict the value.

[0048] Area Criteria – Criteria that holds a value that has a meaning specifically related to a location (a single GPS point, such as a landmark) or a bounded region (a neighborhood, city, etc.).

[0049] Ancestor Value – In a Collapsible Criteria, an Ancestor Value is one that is in the direct ancestor path of a given value (i.e. is a parent, grandparent, etc.).

[0050] Descendant Value - In a Collapsible Criteria, a Descendant Value is one that is in a direct descendant path of a given value (i.e. is a child, grandchild, etc.).

[0051] Criteria Condition – A Boolean expression on the state of current Criteria, where valued, not valued, specifically valued, ancestor and descendant valued can be expressed.

[0052] Context – An identifiable state of the system. Includes the domain of search, Criteria, Data Fields, GUI state, Agent mode, user’s locale, user’s profile and the interaction between the user/client application and system/Agent including what the user has said and the Agent has responded (Conversation Context).

[0053] Active Context – The current system context. In embodiments, there may be a plurality of Active Contexts, comprising a most recent Active Context and one or more prior Active Context that are still considered active.

[0054] Context List – A representation of current and prior Active Context where the most recent Active Context is considered to be highest priority. The list constitutes a context history

where the prior contexts can age (become less relevant) and, optionally, die (be removed from the list and hence become irrelevant).

[0055] Conversation Context – A specific type of context which refers to the state related to what the user or agent has said. There is an implied history to the Conversation Context (the past affects the future).

[0056] Relevant Context – A matching context condition that is appropriate (relevant) to the current Active Context.

[0057] Resulting Context – The context the system changes to or remains in due to processing of some input.

[0058] Reduction – The process of reducing the number of active candidates of a search. This could include obtaining new conditions that restrict the search space, or more restrictive values of current conditions.

[0059] Relaxation – The process of relaxing the current conditions to allow more active candidates of a search. This could include deleting one or more conditions or replacing one or more with less restrictive values.

[0060] System Process Commands – A set of formal actions that change the state of the current system.

[0061] Genre Mapping – An NLU technique which maps Genre Tagged user input (or simulated user input) to System Process Commands.

[0062] Disambiguate – The act of resolving an ambiguity between two or more possible interpretations of user input, such as requesting the user to choose a particular interpretation when the system is unable to determine the proper one among several ambiguous choices (e.g. which city “Richmond” is intended), or the system using additional information, such as context, to automatically choose the best interpretation. Disambiguate intent can refer to the act of Reduction (the active search candidates are considered the ambiguity).

[0063] The following provides systems and methods providing intelligent context awareness wherein context awareness is automatic, based upon user input type, and performance of an operation based on the determined context awareness. Additionally, systems and methods are provided for contextual understanding of user input for effective and accurate searching of

relevant information. Furthermore, systems and methods are provided for automatic and proactive context awareness, wherein user input in a context initiates a system to respond as well as counter-question the user to intelligently narrow down generic queries to specific ones that lead to obtaining a relevant result.

[0064] Embodiments disclosed recite systems and methods for performing an operation or operations, based on contextual commands, which operations further comprise interactively searching for information wherein the system asks key questions to lead the user to the desired results in as few steps as possible. The system comprises a first computing device (including, but not limited to, personal computers, servers, portable wireless devices, cellular phones, smart phones, PDA's, video game systems, tablets, smart televisions, internet televisions, and any other specialized devices that comprise computing capability), which narrows down what the user is asking for through follow-up questions and answers, wherein a search query is transformed into an interactive list of choices resulting in a short list of appropriate results. Embodiments comprise voice recognition, and also wherein the system simulates a human conversation, receiving voice commands, interacting in context and pro-actively asking appropriate questions to disambiguate the user's original request and obtain the user specific desire to find appropriate results. Alternate embodiments include systems which may receive text input and respond textually, receive text input and respond with voice based output, and receive voice input and respond textually. Other variations are possible as would be apparent to a person having ordinary skill in the art.

[0065] Embodiments comprise a computer automated system for interactively searching for information, comprising a processing unit and a memory element, and having instructions encoded thereon, which instructions cause the system to: receive a voice input command which corresponds to a search that can be performed in a context; return, in response to the voice input command in the context, at least one of a search result and an interactive list of relevant choices; if an interactive list of relevant choices is returned, receive a voice input selection of at least one of the returned choices; and wherein the relevant choices are comprised in dynamically generated real-time interactions based on the input voice commands.

[0066] Embodiments comprise a method for interactively searching for information, comprising: receiving a voice input command which corresponds to a search that can be

performed in a context; returning in response to the voice input command in the context at least one of a search result and an interactive list of relevant choices; if an interactive list of relevant choices is returned, receiving a voice input selection of at least one of the returned choices; and wherein the relevant choices are comprised in dynamically generated real-time interactions based on the input voice commands.

[0067] Embodiments comprise techniques which enable the design and processing of a variety of systems and methods for enabling conversational input textually, in voice, or a combination of both. Embodiments comprise context aware interactive searching and an enhanced user experience, improving usability by guiding the user to desired results by proactively presenting in response to user input, contextually relevant questions when there are too many results/responses returned from a user input query. The contextually relevant questions guide the user to know what kind of information they can provide to find more appropriate content for them (i.e. reduce the list of results). Embodiments comprise programs that determine the best question to ask to reduce the set of results and ultimately reduce the number of question-answer steps to a short list of results. Embodiments enable a shortened development time as the system and method are designed to determine the prompts for information to present to the user, including questions to ask the user, based on the context of user input. Rather than being pre-authored, the appropriate information for which to prompt, including questions to ask, will be dynamically, programmatically calculated/determined based on the current content domain, context and available search results.

[0068] Referring first to FIG. 4, a system in an embodiment is shown. The client 405 comprises an application (app, web app, installed application, etc.) which provides an interface enabling the user to provide input to and obtain output from the system. The client is operable to send a textual, audio or visual input (derived from a keyboard, speech recognition, buttons, selection boxes, gestures, etc.) to the server 410 and receive an output (text, text list, HTML, etc.) to display to the user (by, for example, a monitor or other display). The server 410, comprising a processing unit coupled with a memory element, and having instructions encoded thereon, further includes a plurality of processing units comprising a Natural Language Understanding (NLU) unit 415, a conversation processing unit 435, a command processing unit 440, a criteria manager 420, a search engine 425, a reduction unit 430, a relaxation unit 450, and a response generator 445. A context store 480 may further be provided as a memory store

available to all units of the server enabling them to share context and other information. The units communicate and cooperate with one another to implement the following.

[0069] The NLU unit 415 obtains user input and processes it by a NLP technique to generate a semantic form of the input by segmentation, stemming parts of speech and genrization. It maps semantic form to intent based upon current context of the system and determines whether a request is to be processed through the command processor (system commands), conversation processor (simple and multi-step interactions to acquire information) or directly updates criteria by the criteria manager. The NLU unit 415 is operable to receive a natural language text input from a human being, or an encoded representation of a command (from a GUI), process the input using a NLP technique, and determine whether the input is a system process command (e.g., start over, go back), conversation, or a single/compound request to modify the current criteria (search state). The conversation processing unit 435 is operable to manage one (small talk) or more (conversation) input/prompt sequences which enable the system to provide simple answers, or a complex conversational interaction to answer questions, or determine a criteria change based on complex conditions. The command processing unit 440 is operable to receive requests for process commands (go back, start over, etc.) which may change search state (back in history, start over), generate an interpretation of current results (details, map), or service a request of the client (go to a different domain, give me more results). The criteria manager 420 is operable to maintain the current search state of the system as well as a history of changes. The criteria manager 420 issues requests to the search engine 425 to use current criteria to move the system toward its configured goal.

[0070] The search engine 425 is operable to generate a request on the external search CGI based on the current state of the system. It generates search syntax from current criteria to send to external search CGI, informs the client 205 via response generator 445 of the nature of search to occur, executes requests to search CGI and calls upon the response generator to output search results. The reduction unit 430, which is caused to operate when the results count from the search engine 425 exceeds a configured target, applies content defined and automatic mechanisms to prompt the user to input more specific criteria to narrow down the search and produce intelligent, relevant results. The relaxation unit 450 operates when no (or few) results are found by the search engine 425 and enables content defined and automatic mechanisms to adjust the search criteria in an attempt to find results (e.g. expand search radius). The response generator 445

combines search results. It obtains information from the other units and converts that information to a form for output to the client 405. It may apply output formatters 475 to convert client-independent information into client-dependent and language-dependent requests. The output may comprise results, prompts for input, lists of possible further input, search status information, text-to-speech forms of any of the foregoing and updates to GUI state required of the client.

[0071] The search CGI 455 provides a virtualization of one or more external search APIs 460 in a consistent and standardized manner to the server. A single external data source can be queried using the specific application program interface (API). The output formatters obtain a standardized form of results, lists, etc. and generate an output for a particular domain, language, and client.

[0072] In a more specific embodiment, the system comprises a processing unit coupled with a memory element, and having instructions encoded thereon, which instructions are written with minimal language dependencies. The few language dependencies are isolated into self-contained modules (DLL). The heuristics used are all designed to work regardless of the input or output language, or the locale. As such, extending support to new languages and territories is relatively simple, as would be apparent to a person having ordinary skill in the art.

[0073] In a further embodiment, the NLU unit 415 is operable to differentiate user input between small talk (simple query/response), conversational response (based on conversation context), control commands (user requests to specifically change the state of the app or system), content commands (e.g. requests to change search domain, show map, send related email/tweet etc.), and list selection (textual/verbal input identifying a list item). Additionally, in embodiments, the NLU can receive compound requests to change search state wherein content can be designed to manage simple change requests, which can then be input as a compound statement. For example, “I want cheap Italian near the airport” input by the user is handled by the system as separate requests based on “cheap” (cost), “Italian” (cuisine) and “airport” (search area).

[0074] The system is operable to determine context of user input to determine a next prompt. Context may, in this instance, refer to the current state of the system (e.g. mode), what is known (e.g. Criteria), and what has been said (Conversational context). In an embodiment, the system is operable to temporarily detour through a small talk or conversation and return to continue the

main flow. For example, the reduction unit 430 may inform outputs and/or the response generator 445 may generate outputs. For example, if the user's small talk does not match the most recent Active Context, the response generator 445 may be configured to respond on the basis of a next most recent Active Context for which a response does match the small talk. During the small talk, the matching Active Context may become the most recent Active Context, while the previously most recent Active Context is still active but not as recent. When the user returns to the main flow, the still active but not as recent Active Context may return as the most recent Active Context. In an example:

- i. Agent: What type of Cuisine would you like?
- ii. User: What time is it?
- iii. Agent: It is currently 2:30 pm.
- iv. (optional) Agent: What type of Cuisine would you like?
- v. User: What kind of Cuisine can I choose?
- vi. Agent: The list shows the currently available. You can select from the list, or just say one of them.
- vii. User: OK Italian

[0075] In another embodiment, the criteria manager 420 is operable to search multiple values for specific criteria as union or intersection. For example, if a user is searching for a restaurant that serves pizzas, but is also open to the idea of Buffalo wings (say), then the user can input a request such as "pizza or wings" wherein either result returned is good for the user (the union of the results for pizza and for wings). Alternatively and additionally, say the user is looking for a restaurant that serves burgers and steak, a request such as "burgers and steak" will return results of only those restaurants that serve both burgers and steak (the intersection of the results for burgers and for steak). In these examples, the criteria being searched may be tagged for whether multiple values represent a union (e.g., by the presence or inference of an "or" statement) or an intersection (e.g., by the presence or inference of an "and" statement).

[0076] In another embodiment, the system is operable to recognize user input and search based on excluded criteria. Recognition may be implemented, for example, by operation of the NLU unit 415, conversation processing unit 425 and criteria manager 420, while search may be

implemented, for example, by the search engine 425 operating in cooperation with the search CGI 455. For example, say a user is looking for a restaurant that serves Japanese food, but is particularly not interested in sushi. A request such as “Japanese but not sushi” will yield results of only those Japanese restaurants that don’t serve sushi. For such a request, the criteria being searched includes the term “not”, which the criteria manager may recognize as being a “NOT” operator.

[0077] Further, given a large result set, the reduction unit 430 may provide reduction processing by providing a “smart prompt” to the user for selecting alternate search criteria. A content guided approach in an embodiment may include a domain content developer guiding the system based on current criteria and other context. In an automated system, the reduction unit 430 can determine the best subsequent criteria to collect based on the distribution of results among all the remaining criteria. For example, a list can be presented to users that only contain the items active given the current context (criteria, etc.), for example, the available price levels for top-rated Italian restaurants on the waterfront. In such cases, the reduction unit 430 may apply restriction (replace a criteria value) as well as collection (get currently unvalued criteria). Some criteria have a natural order that provides more to less or less to more restriction on results (e.g. search radius, minimum rating, and price levels). The reduction unit 430 can signal to the system to prompt for one of these criteria, automatically restricting the presented list to those values that will result in a reduction in search candidates.

[0078] Additionally, the relaxation unit 450 may provide relaxation processing, which can be understood as the opposite of reduction processing. This is particularly beneficial where the user provides context that has no search results. In such an instance, the relaxation unit 450 may relax criteria to expand the search results without eliminating important search criteria. In one embodiment, the relaxation occurs automatically wherein the relaxation unit 450 determines which criteria to relax and still obtain contextually relevant results. Alternatively, the relaxation may be content guided, either automatic or user aided wherein the user is asked to modify the content of their request in order to obtain a relevant result. A content guided approach enables a domain content developer to guide the system based on current criteria and other context; in an automated approach, the system is operable to determine the best subsequent criteria to collect based on the distribution of results among all the remaining criteria; and a user aided approach

analyses user queries and based on the queried values returns a list to the user(s) that only contain the items active given the current context (criteria etc.).

[0079] In embodiments, search schema (criteria and their values) are defined for each domain that are independent of language and any underlying search engine, providing standardized schema. External search CGI support access to one or more (mash-up) external search engines and return a result schema (result fields and their values) to the system.

[0080] In embodiments, the response generator 445 provides templates and external output formatters, wherein externally defined CGI are operable to generate appropriate layout of such things as candidate lists, for a particular client target. The output of these formatters, as well as natural human text forms of criteria or result field values can be used in a set of standard output templates defined which can target multiple zones of a client GUI. For example,

- i. Agent Says – Prompts and description of what is being presented or requested
- ii. Status – The current search state
- iii. Info – A list of candidates, details, map, etc.

[0081] In further embodiments, the response generator 445 is operable to respond to user input by performing a context aware search and return a result by reduction, relaxation, and location handling. Preferably, embodiments enable and allow a context awareness wherein an operation can further be performed, upon user selection, in a particular context. Ideal embodiments enable automatic context awareness, and performing an operation based on the context awareness. Additionally, embodiments can feature non-contextual objective, contextual and multiple contextual understanding of user input for effective and accurate searching of relevant information. Preferred embodiments include a reduction method of dynamically and automatically choosing the best criteria to ask the user based on the current search results in presenting a list of possible answers (criteria values) to help the user answer. Preferably, embodiments enable relaxing the criteria automatically where appropriate, in order to get an approximate result when an exact answer/result is not found. Preferably, embodiments include disambiguating addresses and locations where there are conflicts and intelligently understanding relationships within addresses.

[0082] The server 410 is operable to provide context aware interactive search (CAIS), comprising: the client 405 receiving an input of a data item from a user in a first context and providing the data item to the NLU unit 415; NLU Unit 415 in cooperation with conversation processing unit 435, command processing unit 440 and criteria manager 420 performing an operation in the context of the received input; search engine 425 in cooperation with reduction unit 430 reducing a set of results obtained by programmatically determining and, by the response generator 445, returning context relevant questions; or by the NLU unit 415 in cooperation with conversation processing unit 435 and response generator 445 disambiguating the user input (what the user has said) to find the most appropriate short list of results for a specific user input (request). In an embodiment, context includes: a. Criteria, b. Agent or System state, and c. Conversation context. Criteria further comprise normalized values for search criteria determined by the system, from the user, through free input and interaction of the user with the system. This can be provided by conversation processing unit 435 in cooperation with command processing unit 440 and criteria manager 420. Agent (system) state comprises the contextual relevance of a returned result by the system in response to user input (a list, details, map, route, etc.). Conversation context comprises context in respect of the interaction already occurred between the user and the Agent (System). The system comprises a processing unit coupled with a memory element, and having instructions encoded thereon wherein the instructions further allow and cause the system to: recognize context by its relevance, and further to calculate relevance by most recent use. In an embodiment, the system is caused to list active context in most recently used order and the instructions will cause the system to consider the first listed context as the most relevant. In such embodiments relevance of conversational context changes frequently, and can become less relevant (i.e. ages and dies) over time.

[0083] Embodiments recognize general context by its relevance. For example, in respect of user input that returns a set of ambiguous matches, the most relevant context is the context in which the input was most recently used. And thus that most recently used context is applied in returning a result. So, for a set of ambiguous matches, that associated with the most recent context would win. Context can also include settings such as user preferences, user location, and user language. Further, conversational context is also recognized in a user interaction and the recognition evolves as the conversation progresses. In an embodiment, contextual relevance determination comprises maintaining a priority list (descending order of priority) of conversation

contexts (a conversation context list or CC List); each with an attribute of some abstract time the context was visited, and applying a pop-to-front methodology. For example, the abstract time could be actual time, or an interaction number. A conversation context is defined as $C_n(t)$, where n is the context number and t is its time attribute. For example, say there are three conversation contexts C_1 , C_2 , and C_3 . Say Conversation contexts C_1 , C_2 , and C_3 are first visited in order, one after the other, and interaction number is the “time” attribute. Thus,

CCList1: $C_1(1)$

CCList2: $C_2(2)$, $C_1(1)$

CCList3: $C_3(3)$, $C_2(2)$, $C_1(1)$

[0084] As shown above, C_3 is more relevant than C_2 , which in turn is more relevant than C_1 . However, if at time $t=4$, Context C_1 is revisited, then C_1 regains the highest relevance, and is caused to pop back to the front. Thus, we will have CCList4: $C_1(4)$, $C_3(3)$, $C_2(2)$.

[0085] Contextual relevance determination may further comprise context death. Context death may be defined as the death of a context when it reaches the end of a queue. The server 410 may implement a queue of a fixed (preconfigured) length or may be configured to dynamically define a queue based on usage and other variables, as defined, for example, by the content developer. Using a fixed queue length, and/or explicit aging and a context lifetime wherein for example in a fixed length of three, C_4 is visited. Thus we have CCList5: $C_4(5)$, $C_1(4)$, $C_3(3)$ which cause C_2 to fall off the end of the queue and die. Essentially, the system may be pre-programmed to keep contexts alive only for three interactions, in this example. So, when C_1 is revisited, and then C_4 , the queue comprises:

CCList6: $C_1(6)$, $C_4(5)$, $C_3(3)$

CCList7: $C_4(7)$, $C_1(6)$

[0086] In the above example, which is for illustrative purposes only, C_3 dies because the current time 7 minus C_3 ’s time 3 is greater than 3. In practice, the list length is likely to be much longer than 3 and the lifetime of the contexts may be varied depending on the nature of the contexts. Additional modifications are possible as would be apparent to a person having ordinary skill in the art.

[0087] In a further embodiment, the system is operable to be configured to provide a dynamic, continuously evolving interactive capability. The system comprises a hybrid automated and rule-based agent comprising a processing unit and memory element, and having instructions encoded thereon, which instructions cause the system to develop evolving interactive agent (system) capability without having to author scenarios for each user interaction (i.e., enabling a developer to create an intelligent, automated interaction system which determines an interaction based on the context and content). The instructions further cause the system to define rules to be applied to the NLU unit 415, which enhances the automated functionality and implements NLP, which comprises mapping of user input to meaning. NLP may further comprise genre tagging, which comprises matching of words and phrases of user input to a normalized semantic form for comparison with content. Genre tagging may further comprise applying (analyzing) parts of speech from a morphological analyzer to address ambiguous genre tagging. For example, the system may be operable to differentiate between “set” the noun and “set” the verb. Additionally, encoded instructions may cause the system to create a hierarchical structure for allowing matching to more and more general ancestors. Additionally or alternatively, NLP may further comprise automatic conversion of a string in a natural language to a structured form which provides a basis for determining meaning (semantics). Representative techniques may comprise: word tokenization, implemented for languages like Chinese and Japanese, for example, which do not have space separation for words; morphological analysis, which entails determining parts of speech, i.e. verb, noun, adverb, etc.; and dialogue act, which is an indicator of the nature of the sentence as a whole (question about location, statement of desire, etc.). In one embodiment, NLP extends these techniques to comprise processing based on context, and genre tagging.

[0088] The system is further operable to generate a simple string representation for easy matching to content. A genre is a representation of a semantic concept consisting of three parts: (a) a normal, which is a canonical (normalized) representation of a potentially large set of synonyms/phrases/sentence fragments (perhaps in multiple languages), (b) a family, which is a grouping of associated normals, and (c) the raw word or phrase from user input associated with the genre. The genre may be represented by a data structure or a string. For purposes of simplicity, the genre may be represented as the string of the form Normal_Family_Raw. Content can define a set of words and phrases that are to represent the semantic concept of a particular Normal_Family. For example:

Italy_Cuisine = Italian cuisine, Italian food, some food from Italy, Italian

FiveStar_Rating = five star, 5 star, the best

Remove_Action = delete, remove, eliminate, take away

[0089] The system may then replace user input with a form which contains genres, which may be referred to as “genre tagging”, or simply “tagging”. An example follows:

User input: The best Italian food

Tagging: FiveStar_Rating_the%20best Italy_Cuisine_Italian%20food

[0090] The raw user input is thus tagged with associated genres. To make things easier to read, the raw user input portion can be left off, as follows:

Tagging: Italy_Cuisine FiveStar_Rating

[0091] The system may apply dynamic normalization in instances where extremely useful families have a set of possible normal that is too large to be feasible to define in content, such as Numbers, Time, Date, etc. For example, it may be useful to deal with time in the following manner: if the user inputs a time, it can be placed in the criteria titled StartTime. This can be accomplished by defining a genre mapping rule that uses the family of a dynamically normalized genre: _Time → Set a StartTime criteria to the value associated with the normal. Dynamic normalization refers to the ability to dynamically (at run-time) create the normal for the genre. For example,

User input: 1:32pm

Tagged form: T1332_Time

[0092] The T1332 is a dynamically created normal. This may be accomplished by defining a matching grammar that matches the user input, captures information in that input, and then passes that information to a specific conversion routine (potentially content defined) to create the normal from the captured data. Then the content developer can define a genre mapping rule for dealing with all-time input:

Matching Condition: _Time

Operation: Set the StartTime criteria to the value associated with the Normal of Time. For example T1332 → StartTime = 13:32

[0093] Genre mapping is a NLU method of mapping the genre tagged form of user input (syntax) to rules for handling that input (semantics). The system matches the user input against genre mapping rules, and consumes the associated parts of the tagged input as the rules are applied. A single genre mapping rule definition may comprise:

- a matching condition, which may either be a Boolean expression of matching genre, wherein associated with this is an optional list of key matching genre for purposes of applying the operation, or a matching grammar, where key matching genre is that genre of the matching genre expression that is used to extract information, specifically the normal;
- optional relevant contexts such as a Boolean expression of currently defined criteria, agent/system state (e.g. showing a map, showing details, etc.), conversation context, user preferences, locale, language, etc.;
- one or more operations to perform, for example, set criteria, add to criteria, delete a value, present a list of values of criteria, send an email, show a map, output a message, etc.;
- associated genres, which are other genres that if present represent the same semantics, and should be consumed with the processing of the Genre Mapping.

[0094] Matching genre forms are a representation of genre for purposes of matching to genre tagged representations of input. They may comprise: (a) an optional normal, (b) the family, or (c) a raw keyword representation. For example, these may be represented as Normal_Family or _Family (any normal of the family) and Raw (specific keyword match).

Normal_Family_Raw matches to Normal_Family

Normal_Family_Raw matches to _Family

Normal_Family_Raw matches to Raw

[0095] This is a Boolean expression of matching genre and allows complicated matches against the user input. Given matching genres A, B and C, Boolean expressions such as the

following can be defined:

- A: matches if A occurs anywhere in the remaining user input
- A and (B or C): matches if and only if remaining user input matches A and either/both B or C
- A and not B: matches if and only if remaining user input matches A but not B

[0096] The orders the genres appear in the user input versus the matching condition definition may be disregarded, as may the presence of intervening other genres or keywords. Hence, a single rule can be written to handle both these user inputs. For example,

Input: “remove Italian”

Tagging: Remove_Action Italy_Cuisine

Input: “Italian, remove please”

Tagging: Italy_Cuisine Remove_Action (“please” removed as an unimportant word)

[0097] In cases where the order is important, a matching grammar may be used instead of a matching condition.

[0098] Alternately, a Boolean expression of currently defined criteria may be applied. This is a relevant context expression that is a Boolean expression on currently defined criteria. This allows content to define genre mappings that only match if certain criteria are defined or not defined. For example, for criteria X, Y and Z:

- X has a value and (Y has a value or Z has a value)
- X has a particular value and Y is not a particular value

[0099] An agent/system may define many genre mapping rules for handling user input in the particular domain of the agent. Content can be used to define a genre mapping rule wherein in response to user input for (say) a restaurant serving a particular cuisine, a rule may be executed which sets a search criterion of food type to the user input cuisine asked/searched for. Or (say) a user is looking for a local business of a particular type, the search criterion may be set accordingly. For example, if Italy_Cuisine is input by the user, then a rule may be executed

which sets a search criterion Food Type to Italian. The following indicates an exemplary system response to such an example of user input:

Matching Condition: Italy_Cuisine

Operation: Set FoodType criterion to the FoodTypeValue Italian

[00100] An even more powerful abstraction is possible in that the genre mapping may request a match of the FAMILY Cuisine, then assigns the FoodType value associated with the normal. So in that case the following may occur:

Matching Condition: _Cuisine

Operation: Set FoodType criteria to the FoodTypeValue associated with the normal of the _Cuisine tagging of the user input.

[00101] Additionally, as genre mapping matches rules and processes them, it removes the taggings that were matched, and then continues to see if there are other rules to apply.

Initial tagging: Italy_Cuisine FiveStar_Rating

Matching Condition: _Cuisine

Operation: Set FoodType criteria to the FoodTypeValue associated with the normal of the _Cuisine tagging of the user input.

Result: Cuisine = Italian

Remaining tagging: FiveStar_Rating

Matching Condition: _Rating

Operation: → Set RatingLevel criteria to the RatingLevelValue associated with the Normal of the Genre matching _Rating.

Result: RatingLevel = 5

Remaining tagging: none

[00102] Also, if the user were to input multiple instances of the same genre, these may also be automatically handled, as follows:

Input: "Italian or pizza"

Initial Tagging: Italy_Cuisine Or_Relation Pizza_Cuisine

Matching Condition: _Cuisine

Operation: Set FoodType criteria to the FoodTypeValues associated with the normal of the _Cuisine tagging of the user input

Result: FoodType = Italian or Pizza

[00103] Another possible form of a matching condition is a grammar, which is a sentence or sentence fragment using matching genre forms that is matched against currently remaining user input, and must match fully and in order. An example is:

Tagging: Remove_Action Italy_Cuisine Add_Action France_Cuisine

[00104] Note that a simple Matching Condition would lead to wrong behavior:

Initial tagging: Remove_Action Italy_Cuisine Add_Action France_Cuisine

Matching Condition: Add_Action _Cuisine

Operation: Set FoodType criteria to the FoodTypeValues associated with the Normal of the _Cuisine tagging of the user input.

Result: FoodType = Italian or French

[00105] However, the system may define grammars as:

Initial context: FoodType = Italian

Matching Grammar: “Remove_Action _Cuisine”

Operation: Remove FoodTypeValue associated with the Normal of the _Cuisine from the Food Result: FoodType = <none>

Matching Grammar: “Add_Action _Cuisine”

Operation: Set FoodType criteria to the FoodTypeValues associated with the Normal of the _Cuisine tagging of the user input

Result: FoodType = French

[00106] An embodiment of an automated and rule-based agent/system in accordance with the above comprises a protocol which may be referred to as NPCQL (NetPeople Content Query

Language) which defines a method for querying and obtaining results from a Content Provider optimized for the context aware interactive search. This enables access to a fixed data schema simplifying content access to mashups. Preferably, NPCQL is provided such that there is no dependency on any one API or content provider and comprises means to separate specific content Provider API calls from specific requests and results returned, as described in the disclosed embodiments. NPCQL enables the agent/system to access 3rd party content without any dependency on the content provider itself. Thus content providers can be changed and added (“mashed up”) without any changes required to the agent/system. Alternatively, 3rd parties can integrate with the system by simply supporting the NPCQL protocol. Additionally, NPCQL comprises defined data schema for each content domain. For example, Restaurant search will have a schema for criteria and result data standard for restaurant search such as Food Type, Service Types, Budget, etc. This schema can be easily added to without affecting existing implementations. Schema used for specific domains will incorporate generic data such as time and budget, with specific data such as Food Type.

[00107] The server 410 may provide auto-disambiguation by learning. In an embodiment, the NLU unit 435 determines that disambiguation is required, performs disambiguation or directs the conversation processing unit 425 to obtain user assistance, which outputs a prompt to the user via the response generator 445. Preferred embodiments include encoded instructions which allow the system to learn in an automated fashion. For example, ambiguous things can be learned to be not ambiguous in a practical sense from user choices. Say, for example, a user input is “Toronto”. The system must determine whether the user meant Toronto, ON or Toronto, OH. If, for example, 99.9% of people choose Toronto, ON, the system may be programmed to consider the proper semantics of “Toronto” IS Toronto, ON and if the user intends to input Toronto, OH then they will naturally know that they need to be specific (i.e. they need to input Toronto, OH due to the learnt familiarity that most people will interpret an input of just Toronto to mean Toronto, ON). Alternatively, the system may recognize a user pattern, and based on input by an identified user, can understand (say) an input of “Toronto” to mean Toronto, OH. Additionally and alternatively, the system can perform auto-disambiguation based on domain (interaction subject), locale/location (where the user is), gender, language, etc. Auto-disambiguation can be based on many other parameters and on variations of the above mentioned parameters, as would be apparent to a person having ordinary skill in the art.

[00108] Preferred embodiments comprise a plurality of sub-systems interconnected with/to each other, and each specializing in a particular domain. Thus, many agents/systems with a domain of expertise can be queried by a single user input, and return a confidence level for the individual agent's ability to handle the input. The full processing can then be passed to the best handler.

[00109] Preferred embodiments of the system comprise data transformers (e.g., output formatters 475) to transform information for the user into the best display (or audio) format for the target client 405. Data transformers can be used for different clients (e.g. smart phone, tablet, TV, etc.), different domains (Restaurants, Local Businesses, Grocery Stores, etc.), different countries, etc. The existence of data transformers allow the agents to be generic to any device and content they are dealing with and yet provide the best display possible for the user. In a specific example, a data transformer may receive a request from NetPeople to format unformatted content data for a specific device in the specified context of the interaction. The request may contain information to assist in formatting such as the language, area, number of characters permitted, etc. For example, if a list of restaurants is being requested the raw NPCQL data would be provided to the transformer with the device type and context (amongst other relevant information) and the transformer would return a formatted list of restaurant items that can be sent directly to the targeted client for display.

[00110] Various exemplary methods are now provided.

[00111] Referring now to Fig. 1, a process flow in an embodiment of a system that enables searching for content and information through conversational interaction is shown. Embodiments can facilitate voice based as well as textual conversational interaction, though preferred embodiments of NetPeople allow for voice based conversational interaction. In step 110, a user inputs a command, textually or by voice, using the client 405. Step 115 comprises performing natural language analysis of the input command by the NLU unit 415. Step 120 is determination of criteria by the criteria manager 420. Step 125 involves searching for content based upon the determined criteria by the search engine 425. Subsequently, in step 130, the system checks the number of results, and accordingly determines if reduction, by the reduction unit 430, or relaxation, by the relaxation unit 450, is desired or required. The reduction step 135 is implemented if too many results are returned and the user is asked to input more specific criteria. The relaxation step 140 takes place if no results are returned and a search is then performed

based on broader, more generic criteria than that input by the user. Thus based on reduction or/and relaxation an automated search is performed and the most accurate results are presented 145 to the user.

[00112] The reduction unit 430 may provide a configurable value for specific domains of the search, such that if there are too many search results (being more than the configurable value), the server 410 is caused to intelligently ask the user for more information to determine what they really want, so that it can narrow, and thereby reduce the results to a short list. The reduction unit 430 may dynamically and automatically choose the best criteria to ask the user based on the current search results, and cause the response generator 445 to present a list of possible answers (criteria values) to help the user answer. For example, say a user is looking for restaurants in a particular area. The server 410 may respond by asking (say) “What kind of cuisine are you looking for? Italian, Chinese, Vietnamese, Japanese...” and so on. The system will determine which choices (criteria values) exist so that the user never makes a choice that ends in no results. Preferably the system will not automatically ask for Italian if there are no Italian restaurants in the results.

[00113] Additionally, the reduction unit 430 may support hierarchical criteria values to ensure that the lists of choices are always reasonable. If there are too many choices the reduction unit 430 may look for the parents to create a narrowed, reasonable sized choice list. In an example embodiment, say the user is looking for a business. The user inputs a voice command that asks the system “search for a business in my location”. The reduction unit 430 performs reduction and causes the response generator 445 to respond by asking “Which business category would you like? Bank, Government Office ...” and so on. The user responds by saying “Bank”. The reduction unit 430 again performs reduction to work with specific criteria and causes the response generator 445 to ask “Local Bank, Trust Bank...” and so on. Thus the server 410 performs targeted, relevant searches that reduces by narrowing, and thereby in some instances eliminates searching for unnecessary, irrelevant items. However, there are cases where the content developer may want to control the criteria questions. The server 410 enables content rules to be defined and take priority over the automated system rules. Thus, based on log analytics, content rules and criteria are tuned to provide the most natural user experience. Variations and modifications of the above are possible, as would be apparent to a person having ordinary skill in the art.

[00114] Referring now to Fig. 2, an example of reduction is shown. After a user has input search criteria, top results may be presented in step 205. In 210 criteria of all the results presented are determined. In step 215 the best (most relevant) criteria is/are calculated, determined, and picked, wherein the calculation of the best criteria in preferred embodiments results in elimination of most results. Of the remaining (picked) criteria value choices the user is asked to select the best criteria in step 220. Subsequently, in step 225 the system returns with top results based on the reduced selection.

[00115] On the other hand, when no (or few) search results are found then the relaxation unit 450 may relax or broaden the criteria, automatically where appropriate. After relaxing the criteria, a search may be performed and the server 410 may return to the client 405 new search results. For example, say the user is looking for shops within a 1 kilometer radius of his or her current location, and there are none found in the search. The relaxation unit 450 may relax (broaden) the criteria, and the server 410 may perform the search within a 2 kilometer radius (say) and return the following result: “I couldn’t find any shops within 1 km so I have expanded (broadened) to 2 km and found 5 shops. Here they are!” Relaxation rules may be defined in content where appropriate. The following are examples of relaxation rules:

Restaurant

If the user sets the “service” criteria, the system will try to remove it and re-search.

If the proximity is used then the system will try to expand the proximity.

Local Business

If the user defines a special shop/place name and the merchant type, the system will try to remove the merchant type and re-search.

[00116] Referring now to Fig. 3, an example of relaxation is shown. After a search based on user input criteria yields no results, the relaxation unit 450 determines which of the user input criteria to broaden (loosen up) in step 305. The search is then performed again with the determined best criteria value to broaden (loosen) in step 310. Subsequently in step 315, results of the search performed with broadened (loosened) criteria values, are returned and presented to the user.

[00117] The NLU unit 415 may further comprise instructions that cause it to recognize address information, locations, landmarks and station names. Preferably, the server 410 further comprises means to disambiguate addresses and locations when there are conflicts. For example, if a user enters “Oakland” for a search the server 410 can respond with “Did you want Oakland, California or Oakland County, Michigan?” In a preferred embodiment the server 410 may understand the parent-child relationships within addresses (neighborhood to city to state to country), and uses common ancestor (parent, grandparent, etc.) entities to aid in the disambiguation, so that if, for example, the user says “Oakland” and the user is in San Francisco (as determined from a reverse geocode of their GPS coordinates), then the server 410 understands it as Oakland, California, USA. via the relationship of a particular Oakland to California and the context of the user being in California and hence the most obvious intent of the user is their local meaning of “Oakland”. Another example would be a neighborhood “Chinatown” which has many incarnations in various places, but can be disambiguated by a common address with the user (e.g. in the same city). Thus, as shown above, a preferred embodiment server 410 can “understand” the relationships within addresses, so that if the user says “San Francisco” then the system understands it as San Francisco, California, USA as determined from a reverse geocode of their GPS coordinates, and any other relevant criteria. Further, rules are tuned and added based on user log analytics to improve the user experience

[00118] Preferably, the server 410 comprises instructions that enable it to set/add one or more criteria tentatively rather than absolutely, and then automatically remove the setting if the search returns no results. For example:

Absolute Criteria Setting:

Set value Ambience = Fun

Search

Show results (possible none)

Tentative Criteria Setting:

Set value tentatively Ambience = Fun

Search

Case 1: Zero search results—Remove Fun from Ambience and Search Again

Case 2: One or more search results—Tentative setting becomes Absolute setting

Show Results for both cases

[00119] Note, other context may have changed along with the tentative setting(s), so this is not necessarily the same as backing out the last set of changes. Rather, only the sets that were marked as being tentative are backed out.

[00120] While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative and not restrictive of the broad invention and that this invention is not limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art upon studying this disclosure. In an area of technology such as this, where growth is fast and further advancements are not easily foreseen, the disclosed embodiments may be readily modifiable in arrangement and detail as facilitated by enabling technological advancements without departing from the principals of the present disclosure or the scope of the accompanying claims.

CLAIMS

1. A computer automated system adapted to interpret commands contextually, and comprising a processing unit and memory element having instructions encoded thereon, which cause the system to:
 - receive a user input command which corresponds to an instruction for performing an operation in a context;
 - disambiguate the input command;
 - perform the operation based on the disambiguated input command; and
 - return in response to the disambiguated input command, zero or more results.
2. The system of claim 1 wherein the instructions further cause the system to:
 - if zero results are obtained, broaden the user input criteria to cause the system to return one or more alternate results in an updated context;
 - if one or more results are obtained, return the one or more results in the input context; and
 - wherein the one or more results are comprised in dynamically generated real-time interactions based on the disambiguated input context or updated context, respectively.
3. The system of claim 1 wherein the instructions that cause the system to disambiguate the input commands further cause the system to disambiguate at least one of user input intent and user input meaning.
4. The system of claim 3, wherein to disambiguate user input intent further comprises reduction, wherein in a plurality of obtained results, the system is caused to narrow the total number of results by their contextual relevance;
 - wherein the said narrowing is comprised in the dynamically generated real-time interactions, which further comprise automatically calculated responses to the input commands to reduce the set of results and the number of steps.
5. The system of claim 3 wherein in disambiguating user input meaning, the encoded instructions further cause the system to:

in response to user input, return a single or plurality of system interpretations for user clarification of an ambiguous input.

6. The system of claim 1 wherein context includes at least one of:

- a content domain, criteria, data fields, GUI state, system state, user's locale, user's profile and the interaction between the user application and system including user input and returned system response; and
- wherein said criteria comprise normalized values for user intent criteria determined from free input and interaction with the system;
- wherein said system state comprises contextual relevance of returned response to user input; and
- wherein conversation context comprises content exchanged during real-time interaction with the system.

7. The system of claim 6, further comprising instructions that cause the system to: perform the operation, which further comprises searching multiple values for specific criteria as at least one of a union and an intersection.

8. The system of claim 6, further comprising instructions that cause the system to, in response to the input, perform the operation based on excluded criteria in said input, wherein said excluded criteria comprises inputting what not to look for.

9. The system of claim 6 further comprising instructions that cause the system to:

- in respect of a disambiguated input context, determine most relevant criteria to collect based on the distribution of results;
- return a single or plurality of determined criteria collected, for user selection; and
- narrow the returned result by contextual relevance, wherein the narrowing comprises analysing obtained results in the real time interaction, and returning one or more results comprising items active given the current context of the input criteria.

10. The system of claim 1 further comprising instructions that cause the system to:

in response to a result returned from a user input in a first context, receive an input in an unrelated context, and in response to said unrelated context, temporarily detour away from the said first context;

return a result in respect of the said unrelated context; and

revert back to the said first context;

wherein context comprises the current state of the system.

11. The system of claim 1 wherein the instructions further cause the system to obtain contextually relevant results, which comprises:

automatically broadening input criteria when zero results are obtained;

determining by analyzing the results obtained based on broadened input criteria, the most relevant results to collect, which determining is based on the distribution of results among active, broadened criteria; and

returning one or more determined, relevant results.

12. The system of claim 11 wherein the analyzing further comprises:

mapping a user input to meaning which comprises matching words and phrases of the input command to a normalized semantic form for comparison with the content;

creating a hierarchical structure for allowing matching of the input command to at least one of general ancestors and descendants; and

converting a string in natural language into a structured format for determining meaning semantics.

13. The system of claim 1 further comprising instructions to create a normal for a genre that the input content belongs to, wherein creating the normal comprises:

defining a matching grammar that matches the input command;

capturing information in that input; and

passing the captured information to a specific conversion routine to create the normal from the captured data;

wherein the conversion routine is content defined.

14. The system of claim 13 further comprising instructions that cause the system to:
implement genre mapping, which comprises matching user input against genre mapping rules, wherein any tagged input is consumed as the rules are applied.
15. The system of claim 1 further comprising instructions which cause the system to, in response to the input command, query and obtain results from a content provider optimized for a context aware interactive search.
16. The system of claim 1 further comprising instructions which cause the system to, in response to the input command, request an external system to perform an operation based on criteria obtained from a context aware interaction.
17. In a computer automated system adapted to interpret commands contextually, and comprising a processing unit and memory element having instructions encoded thereon, a method comprising:
receiving a user input command which corresponds to an instruction for performing an operation in a context;
disambiguating the input command;
performing the operation based on the disambiguated input command; and
returning in response to the disambiguated input command, zero or more results.
18. The method of claim 17 further comprising:
if zero results are obtained, broadening the user input criteria to cause the system to return a single or plurality of alternate results in an updated context;
if one or more results are obtained, returning the one or more results in the input context; and
wherein the one or more results are comprised in dynamically generated real-time interactions based on the disambiguated input context or updated context, respectively.
19. The method of claim 17 wherein disambiguating the input commands comprises causing the system to disambiguate at least one of user input intent and user input meaning.
20. The method of claim 19 wherein disambiguating user input intent further comprises:

reduction, wherein in a plurality of obtained results, narrowing the total number of results by their contextual relevance;

wherein the said narrowing is comprised in the dynamically generated real-time interactions, which further comprise automatically calculating responses to the input commands to reduce the set of results and the number of steps.

21. The method of claim 19 wherein disambiguating user input meaning comprises:
 - in response to user input, returning a single or plurality of system interpretations for user clarification of an ambiguous input.
22. The method of claim 17 further comprising:
 - determining context which comprises normalizing values for input criteria which values are determined from real-time user input;
 - determining system state based on results returned in response to user input; and
 - determining conversation context based on content exchanged during real-time user interaction; and

wherein the said context includes at least one of a content domain, criteria, data fields, a GUI state, system state, user's locale, user's profile and the interaction between the user application and system including user input and returned system response.
23. The method of claim 22 further comprising, performing an operation comprising searching multiple values for specific criteria as at least one of a union and an intersection.
24. The method of claim 22 further comprising allowing the system to, in response to the input, performing an operation based on input excluded criteria.
25. The method of claim 22 further comprising:
 - in respect of a disambiguated input context, determining most relevant criteria to collect based on the distribution of results;
 - returning a single or plurality of determined criteria collected, for user selection; and

narrowing the returned result by contextual relevance, wherein the narrowing comprises analysing obtained results in the real time interaction, and returning one or more results comprising items active given the current context of the input criteria.

26. The method of claim 17 further comprising analyzing context and content of user input wherein the analyzing comprises:

mapping a user input to meaning which comprises matching words and phrases of the input command to a normalized semantic form for comparison with the content;

creating a hierarchical structure for allowing matching of the input command to at least one of a single or plurality of general ancestors and a single of plurality of descendants; and

converting a string in natural language into a structured format for determining meaning semantics.

27. The method of claim 17 further comprising creating a normal for a genre that the input content belongs to, wherein creating the normal comprises:

defining a matching grammar that matches the input command;

capturing information in that input; and

passing the captured information to a specific conversion routine to create the normal from the captured data;

wherein the conversion routine is content defined.

28. The method of claim 24 further comprising instructions that cause the system to:

implement genre mapping, which comprises matching user input against genre mapping rules, wherein any tagged input is consumed as the rules are applied.

29. The method of claim 17 further comprising, in response to the input, optimizing content obtained from a content provider, which optimization is based on user input context and content relevance.

30. The method of claim 21 further comprising disambiguating of the said input meaning based on at least one of context of the input content, user location, gender, and input language.

31. The method of claim 17 further comprising:

in response to a result returned from a user input in a first context, receiving an input in an unrelated context, and in response to said unrelated context, temporarily detouring away from the said first context;
returning a result in respect of the said unrelated context; and
reverting back to the said first context;
wherein context comprises the current state of the system.

32. The method of claim 17 further comprising:

in response to the input returning no result, automatically relaxing criteria from the said input to obtain contextually relevant results;
analyzing the input and based on the values input returning active items given the current context of the input criteria.

33. A system comprising a processing unit and memory element, and having instructions encoded thereon, which instructions cause the system to:

receive an input in a context;
return a result in respect of the received context by at least one of narrowing, broadening, and location handling in respect of the input value;
wherein an operation is performed based upon the context of input criteria;
wherein the narrowing further comprises returning a one or more relevant items based on a current result returned in response to the input, and comprising possible criteria values;
wherein the broadening further comprises, when an exact result is not found, broadening the input criteria automatically, and where appropriate, to obtain a result; and
wherein the location handling further comprises disambiguating addresses and locations where there are conflicts based on an input history, and establishing relationships within addresses based upon an input history.

34. A dynamic, self-evolving, computer automated system comprising a processing unit and a memory element, and having instructions encoded thereon which instructions cause the system to:

develop evolving interactive capability without human authoring of scenarios for each user interaction, wherein said evolving further comprises determining a user interaction based on the context and content;

automatically define rules that enhance the automated functionality;

implement natural language processing wherein said natural language processing comprises mapping a user input to meaning, and which mapping further comprises genre tagging;

differentiate between a set which comprises a grouping and a set which comprises the action of making a change; and

create a hierarchical structure for allowing matching of input to at least one of a single or plurality of general ancestors and a single or plurality of descendants.

35. The system of claim 34 wherein the natural language processing further comprises automatic conversion of a string in a natural language to a structured, machine readable format which provides a basis for determining meaning.

36. The system of claim 34 wherein the said genre tagging further comprises:

matching of words and phrases of user input to a normalized semantic form for comparison with content; and

analyzing parts of speech to disambiguate ambiguous input.

37. In a computer automated system comprising a processing unit and a memory element, and having instructions encoded thereon, a method for dynamic self-evolving of the computer automated system, comprising:

developing evolving interactive capability without human authoring of scenarios for each user interaction, wherein said evolving further comprises determining a user interaction based on the context and content;

automatically defining rules to enhance the automated functionality;

implementing natural language processing wherein said natural language processing further comprises mapping a user input to meaning, and which mapping further comprises genre tagging;

differentiating between a set which comprises a grouping and a set which comprises the action of making a change;

creating a hierarchical structure for allowing matching to at least one of a single or plurality of general ancestors or a single or plurality of descendants.

38. The method of claim 37 wherein the natural language processing further comprises automatic conversion of a string in a natural language to a structured, machine readable format which provides a basis for determining meaning.
39. The method of claim 37 wherein the said genre tagging further comprises:
 - matching of words and phrases of user input to a normalized semantic form for comparison with content; and
 - analyzing parts of speech to disambiguate ambiguous input.
40. The system of claim 1 wherein the command further comprises the instruction to perform a search or an operation in the input context.

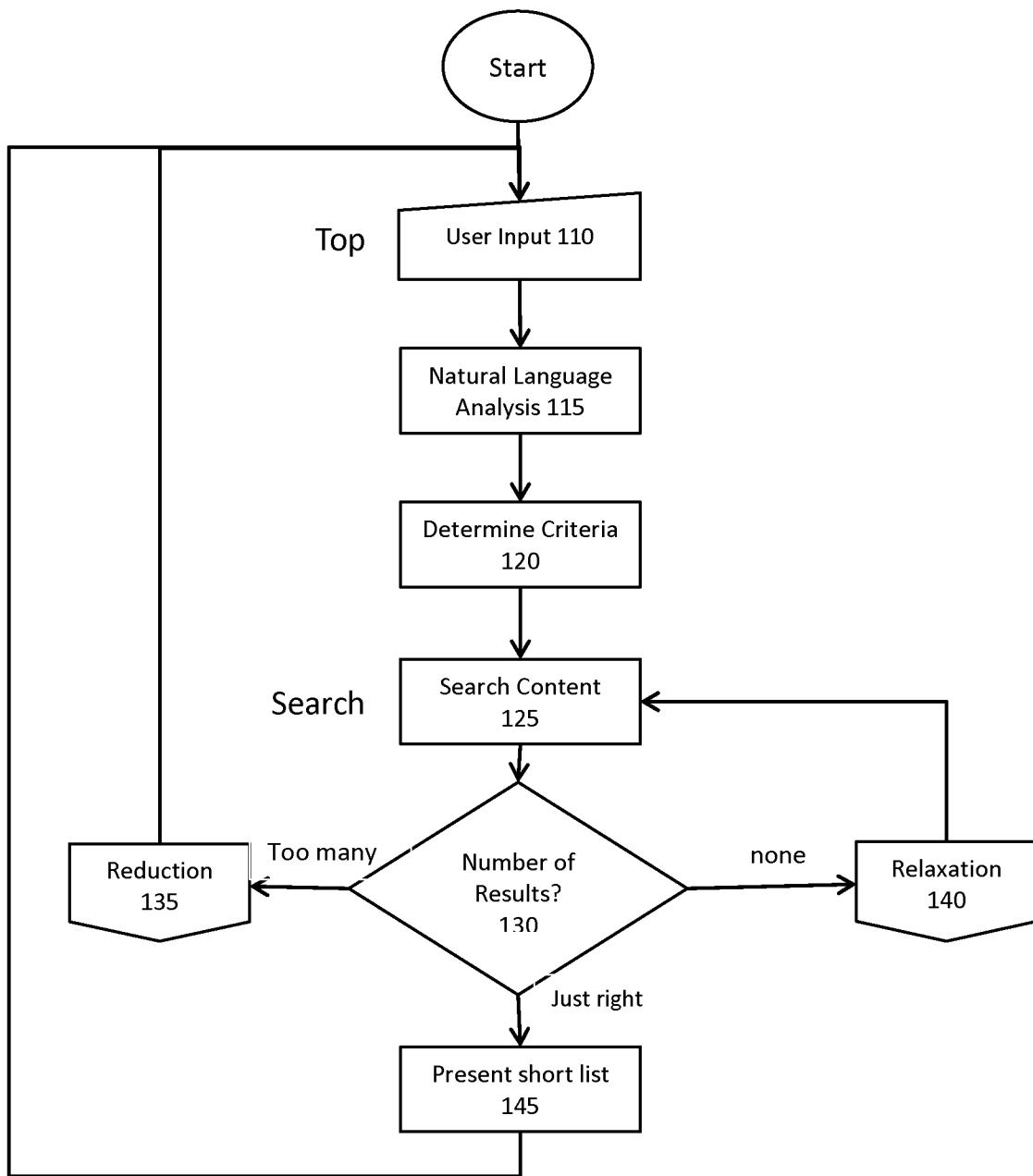
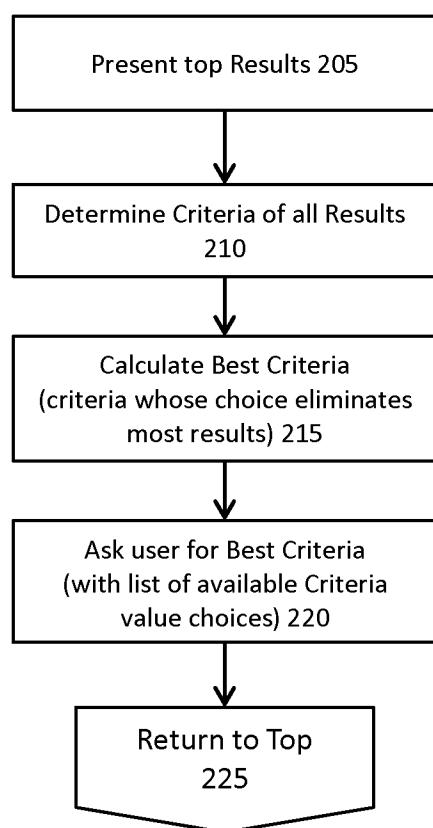
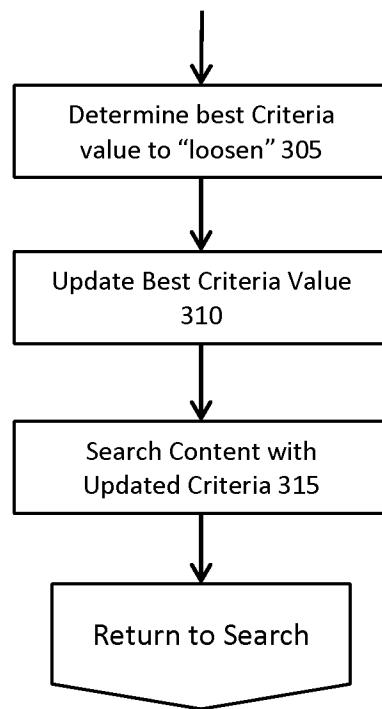


FIG. 1

**FIG. 2**

**FIG. 3**

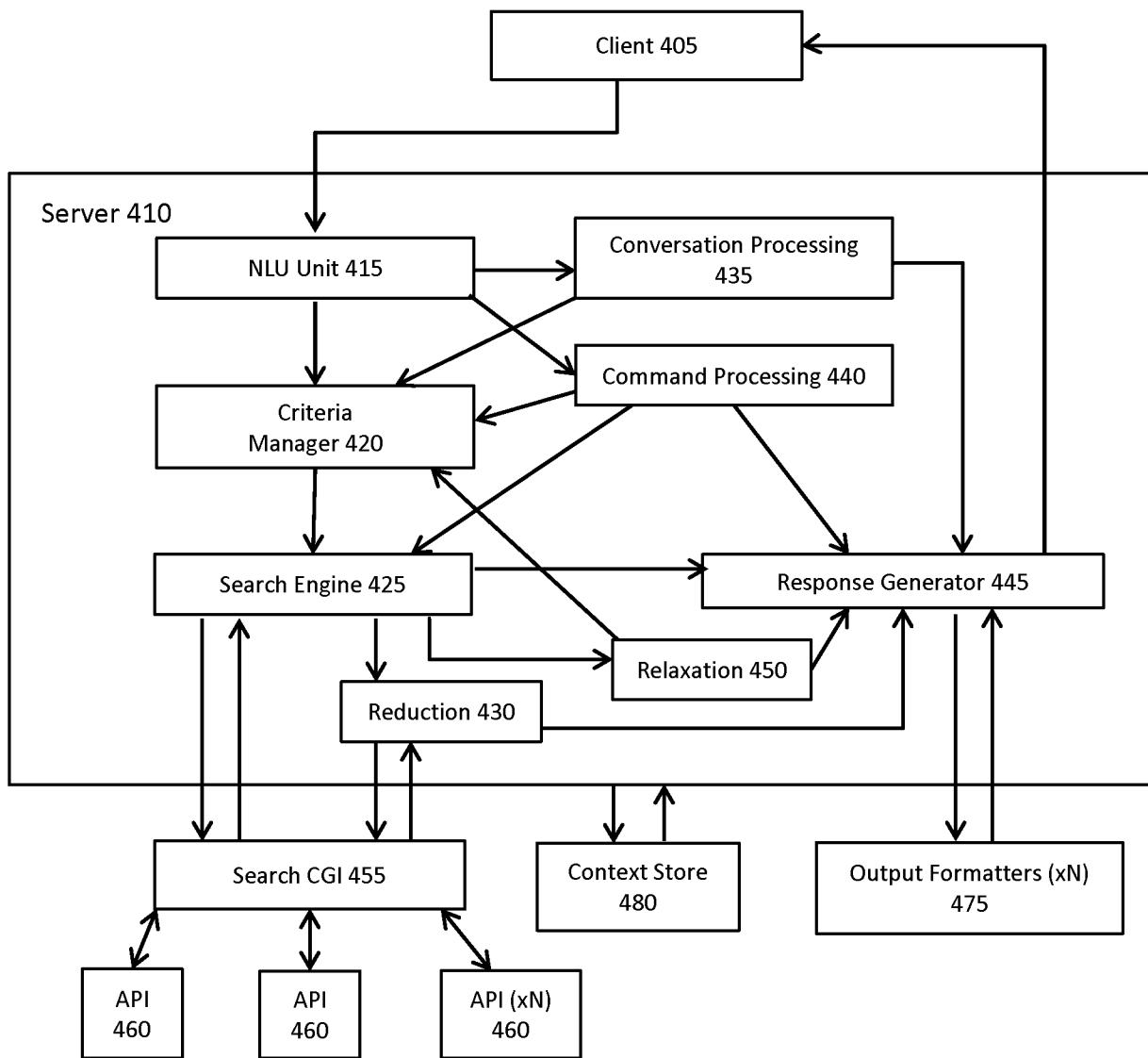


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2013/050181

A. CLASSIFICATION OF SUBJECT MATTER IPC: G06F 17/30 (2006.01), G06F 17/27 (2006.01) According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC (2006.01): G06F 17/30, G06F 17/27		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used) Databases: TotalPatent, Canadian Patent Database, Google		
Keywords: context, disambiguat*, content, result, response, user , input, search, query, command, location, history, hierarchy, parent child, ancestor, descendent, real time, natural, language, automatic, operation, restrict, constrain, narrow, broad, relax, provider, gender, speech		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO2011088053 A2 (GRUBER et al.) 21 July 2011 (21-07-2011) *entire document*	1-40
A	US2007/0061336 A1 (RAMER et al.) 15 March 2007 (15-03-2007) *entire document*	
[] Further documents are listed in the continuation of Box C.		[X] See patent family annex.
<p>* Special categories of cited documents :</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier application or patent but published on or after the international filing date</p> <p>“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)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>		<p>“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</p> <p>“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</p> <p>“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</p> <p>“&” document member of the same patent family</p>
Date of the actual completion of the international search 21 June 2013 (21-06-2013)		Date of mailing of the international search report 26 June 2013 (26-06-2013)
Name and mailing address of the ISA/CA Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476		Authorized officer Tatjana Kremer (819) 934-0312

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2013/050181

Patent Document Cited in Search Report	Publication Date	Restricted Patent Family Member(s)	Publication Date
WO2011088053A2	21 July 2011 (21-07-2011)	AU2010254812A1 AU2011205426A1 CA2787351A1 CA2791791A1 CA2792412A1 CA2792442A1 CA2792570A1 CA2793002A1 CA2793118A1 CA2793248A1 CA2793741A1 CA2793743A1 CN102792320A EP2526511A2 GB2490444A KR20120120316A KR20120136417A KR20120137424A KR20120137425A KR20120137434A KR20120137435A KR20120137440A KR20120138826A KR20120138827A KR20130000423A MX2012008369A US2012016678A1 US2012022872A1 US2012245944A1 US2012265528A1 US2013110505A1 US2013110515A1 US2013110518A1 US2013110519A1 US2013110520A1 US2013111348A1 US2013111487A1 US2013117022A1	12 January 2012 (12-01-2012) 23 August 2012 (23-08-2012) 21 July 2011 (21-07-2011) 21 November 2012 (21-11-2012) 28 November 2012 (28-11-2012) 31 October 2012 (31-10-2012) 01 November 2012 (01-11-2012) 18 December 2012 (18-12-2012) 20 December 2012 (20-12-2012) 20 December 2012 (20-12-2012) 26 December 2012 (26-12-2012) 26 December 2012 (26-12-2012) 02 January 2013 (02-01-2013) 12 September 2012 (12-09-2012) 19 January 2012 (19-01-2012) 26 January 2012 (26-01-2012) 27 September 2012 (27-09-2012) 18 October 2012 (18-10-2012) 02 May 2013 (02-05-2013) 02 May 2013 (02-05-2013) 09 May 2013 (09-05-2013)
US2007061336A1	15 March 2007 (15-03-2007)	US2007061336A1 US7702318B2 US2010211458A1 US7907940B2 US2010198681A1 US7970389B2 US2012010954A1 US8180332B2 US2011258046A1 US8270955B2 US2009240569A1 US2011312310A1 US2012005020A1 US2012130812A1 US2013012178A1	15 March 2007 (15-03-2007) 20 April 2010 (20-04-2010) 19 August 2010 (19-08-2010) 15 March 2011 (15-03-2011) 05 August 2010 (05-08-2010) 28 June 2011 (28-06-2011) 12 January 2012 (12-01-2012) 15 May 2012 (15-05-2012) 20 October 2011 (20-10-2011) 18 September 2012 (18-09-2012) 24 September 2009 (24-09-2009) 22 December 2011 (22-12-2011) 05 January 2012 (05-01-2012) 24 May 2012 (24-05-2012) 10 January 2013 (10-01-2013)