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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 September 2018 (13.09.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/161306 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/CN2017/076133

(22) International Filing Date:
09 March 2017 (09.03.2017)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052 (US).

(72) Inventors; and

(71) Applicants *(for US only)*: **WU, Xianchao** [CN/JP]; One Microsoft Way, Redmond, Washington 98052 (US). **FUJIWARA, Keizo** [JP/JP]; One Microsoft Way, Redmond, Washington 98052 (US).

(74) Agent: **NTD PATENT & TRADEMARK AGENCY LIMITED**; 10th Floor, Block A, Investment Plaza, 27 Jinrongdajie, Xicheng District, Beijing 100033 (CN).

(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, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

(54) Title: APPLICATION RECOMMENDATION

(57) Abstract: A method is provided for application recommendation based on matching rates between a query and applications associated with a chatbot. And a method is also provided for information collection from web data and user data including user chatting log data and user-application usage data. The information collection facilitates the application recommendation in one aspect and provides information for application development in another aspect.

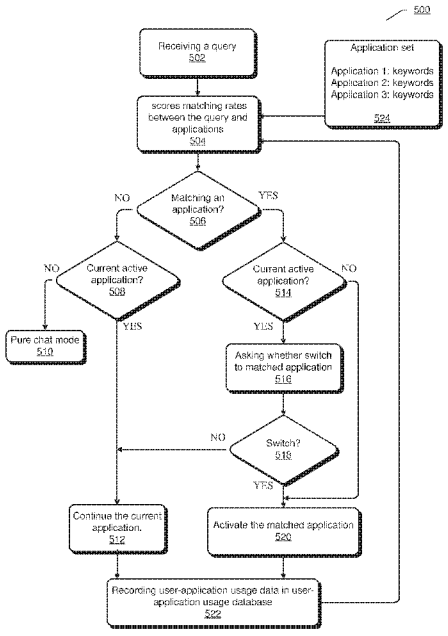

FIG 5

WO 2018/161306 A1

APPLICATION RECOMMENDATION

BACKGROUND

[0001] Artificial intelligence (AI) conversational chat programs are becoming more and more popular. These conversational chat programs, also referred to as "chatbots", allow users to carry on conversations with a virtual entity. In some instances, an application such as a game application, a topic-related application, or the like may be activated during the conversation between a user and a chatbot. As more and more applications are released in the circumstance of the intelligent automated chatting, there may be a large number of applications appended to a chatbot.

SUMMARY

[0002] This Summary is provided to introduce a selection of concepts that are further described below in the Detailed Description. It is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0003] Embodiments of the present disclosure provide a method for application recommendation through intelligent automated chatting. A query is received in a conversation. Matching rates between the query and applications are scored. The query is identified as being matched with a first application of the applications based on the matching rates. The first application is recommended in the conversation.

[0004] Embodiments of the present disclosure provide a method for collecting information. A plurality of user clusters are generated based on user data associated with a service, wherein a user cluster comprises a set of users and a set of key words, and a key word in the user cluster is appended with one or more sentence-level descriptions. The plurality of user clusters are stored in a database.

[0005] It should be appreciated that the above one or more aspects comprise the features hereinafter fully described and particularly pointed out in the claims. The following description and the drawings set forth in detail certain illustrative features of the one or more aspects. These features are only indicative of the various ways in which the principles of various aspects may be employed, and this disclosure is intended to include all such aspects and their equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The disclosed aspects will hereinafter be described in connection with the appended drawings that are provided to illustrate and not to limit the disclosed aspects.

[0007] Figure 1 illustrates an exemplary environment where the described techniques can be implemented according to an embodiment.

[0008] Figure 2 illustrates an exemplary system applying a chatbot according to an embodiment.

[0009] Figure 3 illustrates an exemplary user interface (UI) according to an embodiment.

[0010] Figure 4 illustrates an exemplary conversation between a chatbot and a user according to an embodiment.

[0011] Figure 5 illustrates a flowchart of an exemplary process for application recommendation according to an embodiment.

[0012] Figure 6 illustrates an exemplary user-application usage database according to an embodiment.

[0013] Figure 7 illustrates an exemplary collection of information from user data according to an embodiment.

[0014] Figure 8 illustrates an exemplary collection of information from web data according to an embodiment.

[0015] Figure 9 illustrates an exemplary link of user data and web data according to an embodiment.

[0016] Figure 10 illustrates a flowchart of an exemplary process for updating an application according to an embodiment.

[0017] Figure 11 illustrates a flowchart of an exemplary process for updating key words of an application according to an embodiment.

[0018] Figure 12 illustrates a flowchart of an exemplary process for application recommendation according to an embodiment.

[0019] Figure 13 illustrates a flowchart of an exemplary process for information collection according to an embodiment.

[0020] Figure 14 illustrates an exemplary apparatus for application recommendation according to an embodiment.

[0021] Figure 15 illustrates an exemplary apparatus for information collection according to an embodiment.

[0022] Figure 16 illustrates an exemplary computing system according to an embodiment.

DETAILED DESCRIPTION

[0023] The present disclosure will now be discussed with reference to several exemplary implementations. It is to be understood that these implementations are discussed only for enabling those skilled in the art to better understand and thus implement the embodiments of the present disclosure, rather than suggesting any limitations on the scope of the present disclosure.

[0024] Figure 1 illustrates an exemplary environment where the described techniques can be implemented according to an embodiment.

[0025] In the exemplary environment 100, a network 110 is applied for interconnecting among a terminal device 120, an application server 130 and a chatbot server 140.

[0026] The network 110 may be any type of networks capable of interconnecting network entities. The network 110 may be a single network or a combination of various networks. In terms of coverage range, the network 110 may be a Local Area Network (LAN), a Wide Area Network (WAN), etc. In terms of carrying medium, the network 110 may be a wireline network, a wireless network, etc. In terms of data switching techniques, the network 110 may be a circuit switching network, a packet switching network, etc.

[0027] The terminal device 120 may be any type of computing device capable of connecting to the network 110, assessing servers or websites over the network 110, processing data or signals, etc. For example, the terminal device 120 may be a desktop computer, a laptop, a tablet, a smart phone, etc. Although only one terminal device 120 is shown in Figure 1, it should be appreciated that a different number of terminal devices may connect to the network 110.

[0028] The terminal device 120 may include a chatbot client 122 which may provide a chat service for a user. In some implementations, the chatbot client 122 at the terminal device 120 may be an independent client application corresponding to the chatbot service provided by the chatbot server 140. In some other implementations, the chatbot client 122 at the terminal device 120 may be implemented in a third party application such as a third party instant messaging (IM) application. Examples of the

third party IM message comprise MSN™, ICQ™, SKYPE™, QQ™, WeChat™ and so on.

[0029] The chatbot client 122 communicates with the chatbot server 140. For example, the chatbot client 122 may transmit messages inputted by a user to the chatbot server 140, and receive responses associated with the messages from the chatbot server 140. The chatbot client 122 and the chatbot server 140 may be collectively referred to as a chatbot. As the conversation between the user and the chatbot is performed typically in a query-response manner, the messages inputted by the user are commonly referred to as queries, and the answers outputted by the chatbot are commonly referred to as responses. The query-response pairs may be recorded as user log data. It should be appreciated that, in some implementations, instead of interacting with the chatbot server 140, the chatbot client 122 may also locally generate responses to queries inputted by the player.

[0030] An application 124 may be activated during a conversation between the chatbot and a user. For example, the application 124 may be associated with a trigger word. The user may input the trigger word when the user wants to start the application 124 during the conversation. After receiving the trigger word, the chatbot may activate the application during the conversation.

[0031] In some implementations, the application 124 may be implemented at an application server 130, which may be a third part application server. For example, while the application 124 is active during the conversation, a query from a user is sent to the application server 130 via the chatbot, and a response from the application server 130 is sent to the user via the chatbot. In some other implementations, the application 124 may be implemented at the chatbot server 140, and in this case an application module 142 may be implemented at the chatbot server 140. Applications provided by the chatbot service provider and/or applications provided by third party application providers may be implemented at the application module 142. The chatbot may call an application at the application module 142 in order to activate the application during the conversation.

[0032] It should be appreciated that the application 124 associated with the chatbot service may also be referred to as a feature, a function, an applet, or the like, which is used to satisfy a relatively independent requirement of a user during a machine conversation with the user.

[0033] It should be appreciated that all the network entities shown in Figure 1 are exemplary, and depending on specific application requirements, any other network entities may be involved in the environment 100.

[0034] Figure 2 illustrates an exemplary chatbot system according to an embodiment.

[0035] The system 200 may comprise a user interface (UI) 210. The UI 210 may be implemented at the chatbot client 122, and provide a chat window for interacting between a user and the chatbot.

[0036] Figure 3 illustrates an example of the UI 210. A chat window 320 is displayed on a computing device 300. The chat window 320 comprises a presentation area 322, a control area 324 and an input area 326. The presentation area 322 presents queries and responses in a conversation between a user and a chatbot, which is represented by the icon 310. The control area 324 includes a plurality of virtual buttons for the user to perform message input settings. For example, the user may make a voice input, attach image file, select emoji symbols, and make a short-cut of current screen, and so on through the control area 324. The input area 326 is used for the user to input messages. For example, the user may type text through the input area 326. The control area 324 and the input area 326 may be collectively referred to as input unit. The user may also make a voice call or video conversation with the AI chatbot though the input unit.

[0037] For example, in the UI as shown in Figure 3, the user inputs a message "Rinna, how old are you" as a query, and a message "second year of primary high school" may be output by the chatbot as a response. Similarly, the user inputs a message "do you have breakfast" as a query, and a message "Yes, How about you" may be outputted by the chatbot as a response. Here, Rinna is the name of the AI chatbot, which may also be referred to as AI chat system.

[0038] The queries from the user are transferred to the query queue 232, which temporarily stores users' queries. The users' queries may be in various forms including text, sound, image, video, and so on.

[0039] The core processing module 220 may take the messages or queries in the query queue 232 as its input. In some implements, queries in the queue 232 may be served or responded in first-in-first-out manner.

[0040] The core processing module 220 may invoke processing units in an

application program interface (API) module 250 for processing various forms of messages. The API module 250 may comprise a text processing unit 252, a speech processing unit 254, an image processing unit 256, etc.

[0041] For a text message, the text processing unit 252 may perform text understanding on the text message, and the core processing module 220 may further determine a text response.

[0042] For a speech message, the speech processing unit 254 may perform a speech-to-text conversion on the speech message to obtain text, the text processing unit 252 may perform text understanding on the obtained text, and the core processing module 220 may further determine a text response. If it is determined to provide a response in speech, the speech processing unit 254 may perform a text-to-speech conversion on the text response to generate a corresponding speech response.

[0043] For an image message, the image processing unit 256 may perform image recognition on the image message to generate corresponding text, and the core processing module 220 may further determine a text response. For example, when receiving a dog image from the user, the AI chat system may determine the type and color of the dog and further gives a number of comments, such as "So cute German shepherd! You must love it very much". In some cases, the image processing unit 256 may also be used for obtaining an image response based on the text response.

[0044] Moreover, although not shown in Figure 2, the API module 250 may comprise any other processing units. For example, the API module 250 may comprise a video processing unit for cooperating with the core processing module 220 to process a video message and determine a response. For another example, the API module 250 may comprise a location-based processing unit for supporting location-based services.

[0045] An application recommendation module 280 in the system 200 may determine whether or not to recommend an application to a user in response to a query from the user. Taking the query "Rinna, how old are you" as an example, the application recommendation module 280 may determine that the user does not have an intention to activate an application, and only wants to have a small talk with the chatbot.

[0046] Then, the core processing module 220 may determine a response through an index database 260. The index database 260 may comprise a plurality of index

items that can be retrieved by the core processing module 220 as responses. The index items in the index database 260 may be classified into a question-answer pair index set 262 and a pure chat index set 264. Index items in the question-answer pair index set 262 are in a form of question-answer pairs, and the question-answer pair index set 262 may comprise question-answer pairs associated with an application such as application 124. Index items in the pure chat index set 264 are prepared for free chatting between the user and the chatbot, and may or may not be in a form of question-answer pairs. It should be appreciated that the term question-answer pair may also be referred to as query-response pair or any other suitable terms. Taking the query "Rinna, how old are you" as an example, the core processing module 220 may determine a response "second year of primary high school" through the pure chat index set.

[0047] The responses determined by the core processing module 220 may be provided to a response queue or response cache 234.The responses in the response queue or response cache 234 may be further transferred to the user interface 210 such that the responses can be presented to the user in an proper order.

[0048] A user database 270 in the system 200 is used to record user data occurred in conversations between users and the chatbot. The user database 270 may comprise a user log database 272 and a user-application usage database 274.

[0049] The user log database 272 may be used to record messages occurred in conversations between users and the chatbot. For example, the user log database 272 may be used to record user log data of pure chat. For another example, the user log database 272 may be used to record not only the user log data of pure chat but also user log data occurred while an application is active. The user log data may be in a query-response pair form, or may be in any other suitable form.

[0050] The user-application usage database 274 may be used to store every user's usage information of applications associated with the chatbot or the AI chat service. The user-application usage data in the database 274 may provide information for the application recommendation module 280 to determine how to respond to a user's query and whether or not to recommend an application in response to a user's query.

[0051] Figure 4 illustrates an exemplary chatting flow between a user and a chatbot according to an embodiment.

[0052] In the chatting flow or conversation, a query "Shiritori" is received by the

chatbot from a user. The chatbot, specifically the application recommendation module 280, identifies that the query is matched with an application, and additionally identifies that there is no active application in the conversation currently. Then the Shiritori application is activated, and the chatbot outputs a response to ask the user to choose a normal mode or hard mode for the Shiritori application. It should be appreciated that reasonable modifications may be applicable to this process. For example, instead of directly activate the Shiritori application, the chatbot may output a message asking whether the user wants to activate the Shiritori application even when there is no current active application in the conversation. Either the direct activating of the application or the outputting of the asking message is an example of recommending the application performed by the chatbot.

[0053] When the user inputs a query "Normal", the application recommendation module 280 sends the query to the current active Shiritori application and the conversation proceeds. The user and the chatbot/Shiritori application input "like", "keep", "episode", "decoder", "error", "orange" in turn during the shiritori game.

[0054] A query "I meet a lot with the girl living nearby" is received by the chatbot from the user. The chatbot, specifically the application recommendation module 280, identifies that the query is matched with an application, and identifies that there is an active application in the conversation currently. The chatbot outputs a response for recommending the matched application, that is, "Oh, do you want to do "love diagnosis"?".

[0055] When a positive answer is received, the chatbot switches the current active application "Shiritori" to the recommended application "love diagnosis". A sentiment analysis module may be used to make a positive, negative or neutral judgement of user's query based on user's natural language style inputs. In some implementations, the chatbot ends the current active application, that is, "We played 3 rounds of Shiritori. See you next time!", and activates the matched application, as shown in the Figure 4. In some other implementations, the chatbot may suspend the current active application while the newly matched application is active, and may resume the suspended application when the newly matched application is ended.

[0056] Figure 5 illustrates an exemplary process for recommending an application according to an embodiment.

[0057] The process 500 starts from 502, at which a chatbot receives a query from

a user. The query may also be received from an entity such as an intelligent system or an intelligent bot. From example, the query may be "Shiritori" or any message inputted from the user such as "normal", "like", "I meet a lot with the girl living nearby", or the like, as shown in Figure 4.

[0058] At 504, the chatbot, specifically the application recommendation module 280, scores the matching rates between the query and applications associated with the chatbot. The matching rates between the query and applications may also be referred to as similarities between the query and applications. The two terms matching rate and similarity are interchangeably used herein.

[0059] In some implementations, the application recommendation module 280 is implemented as a machine learning module, which is trained by using user queries, trigger data of applications, user usage data of the applications, and optionally other suitable data.

[0060] For a set of applications 524 associated with the chatbot, each application has a set of keywords which are examples of trigger data. In some implementations, as shown by the arrows from blocks 524 and 522, the application recommendation module 280 computes the similarity scores based on the query, the trigger data of the applications and the user-application usage data stored in the user-application usage database 274.

[0061] At 506, the application recommendation module 280 identifies whether the query is matched with one of the applications based on the scores.

[0062] In some implementations, a threshold may be set for an application, and a matching decision may be made when the similarity score of the application is above the threshold. On the other hand, a not-matching decision may be made when the similarity score of the application is under the threshold.

[0063] In some implementations, if more than one matching application is identified, the application recommendation module 280 may choose the application having the highest score to recommend.

[0064] In some implementations, the threshold of an application may be updated dynamically. Initially, the threshold may be set to be a relatively small value to obtain a better trigger rate of an application. During the usage of the application recommendation, the threshold may be improved to balance the precision and the trigger rate of application recommendation. An exemplary process of updating a

threshold of an application is shown as the following:

1. Initialize the threshold of the application;
2. Set success usage number of the application S to be 0, and failed usage number of the application F to be 0;
3. If the similarity score of the application is larger than the threshold:
    a. Recommend the application;
    b. If the application is finished without interruption and the usage time of the application is above the average usage time of the application, set S = S + 1;
    c. Otherwise, set F = F + 1
4. Update the threshold periodically (such as per week)
    a. precision = S/(S+F), trigger rate = S/number of queries during the period
    b. if precision < expected precision value:
        i. threshold = threshold + alpha (e.g., alpha is set to be a float in the range of (0, 0.1))
    c. otherwise if trigger rate < expected trigger rate value:
        i. threshold = threshold – beta (e.g., beta is set to be a float in the range of (0, 0.1)).

[0065] The expected precision value and the expected trigger rate value may be a predefined threshold. It should be appreciated the updating algorithm is for sake of illustration without any intention to limit the scope of the disclosure.

[0066] When a not-matching decision is made at 506, the process goes to 508, at which it is determined whether a current application is active during the conversation between the chatbot and the user.

[0067] In some implementations, in order to detect whether or not an application is already active, the user-feature usage database is used to store every user's usage information of the set of applications. By checking the user-feature usage database as shown in Figure 6, it is identified that there is no active application currently if the active data 620 is empty, and it is identified that there is an active application currently if the active data 620 is not empty.

[0068] When it is identified that there is no active application currently at 508, the

process goes to 510 at which a pure chat mode is started.

[0069] When it is identified that there is an active application currently at 508, the process goes to 512 and the current active application continues.

[0070] When a matching decision is made at 506, the process goes to 514, at which it is identified whether a current application is active during the conversation between the chatbot and the user.

[0071] When it is identified that there is no active application currently at 514, the process goes to 520 at which the matched application is activated to become the current active application.

[0072] When it is identified that there is an active application currently at 514, the process goes to 516 at which a recommendation for the matched application is outputted. The recommendation may be presented as a question to ask whether the user wants to switch to the matched application.

[0073] If a negative answer to the recommendation is received from the user at 518, the process goes to 512 and the current active application continues.

[0074] If a positive answer to the recommendation is received from the user at 518, the process goes to 520 at which the matched application is activated to become the current active application.

[0075] At 522, the user's usage data of the current active application is recorded in the active data 620 of the user-application data base. When the current active application is ended the active data 620 may be stored as the history data 610.

[0076] Figure 6 illustrates an exemplary user-application usage database 600. There are two types of data in the database, one is log-style data called history data 610 that stores the history usage of all applications of all users, the other is active-style data called active data 620 which records the current active application's statistical information for a specific user, which is identified by a user ID. When the current active application is ended, either by the user or by a timeout exception, the active data 620 may be stored in the history data unit 610.

[0077] The schema of the history data and the active data are identical as shown in 630 and 640, the difference is that some information in the active data is in updating since the active application is still working. For example, the timestamp end information in the active data unit is not available until the current active application is ended.

[0078] An exemplary schema of the user-application usage data is illustrated in 630. The exemplary schema includes user ID, application ID, timestamp start, timestamp end, user query list, application statistics, application session data, is third-party application, application owner. It should be appreciated that more or less information elements may be applicable, and the disclosure is not limited to the specific schema 630. The timestamp start and timestamp end indicate the start time and end time of the application (identified by application ID) used by the user (identified by user ID). The user query list includes the user's inputted queries for this usage of the application. For example, for a cooking application, the user queries may be "how to cook spring roll", "I want to make sushi" and so on inputted in this usage of the cooking application. The application statistics include statistical information about this usage of the application, such as the number of rounds that the user played with "Shirotori" application. Examples of the application session data include how many times this application was ended in a normal way, how many times this application was ended in an interrupted way (for example, the application is interrupted by another application). The application session data may be computed with respect to a single user and a single application in some implementations, and may also be computed with respect to a single application and all users in some other implementations. The application session data may be updated periodically. The element of is third part application indicates whether the application identified by the application ID is provided by a third party application provider. The application owner indicates the application provider's information, such as the third-party developer's name or ID.

[0079] Based on the user-application usage data in this database, machine learning features may be obtained for training a similarity computing model to compute similarity scores between queries and applications. The similarity computing model, as a machine learning model, is implemented in the application recommendation module 280 or is the application recommendation module 280. The exemplary machine learning features obtained based on the user-application usage data are as the following:

1. The accumulated usage frequency of one application with respect to all applications for current user or all users.
2. The accumulated number of interruptions of one application for current

user or all users.

3. The rank of the usage of one application among all applications for current user or all users.
4. The rank of the interruptions of one application among all applications for current user or all users.
5. Averaged usage time (e.g., in seconds) of one application for current user or all users.
6. Maximum usage time (e.g., in seconds) of one application for current user or all users.
7. Accumulated usage time (e.g., in seconds) of one application for current user or all users.
8. Average time point that one application is triggered by current user. For example, the time point is mainly in the morning for a weather report application, is near 12 am or 6 pm for a food recommendation application.
9. Averaged query length of user's queries for one application. For example, the longer the query is, the more the user is engaged in the current application.
10. Sentiment tendencies of user's queries for one application. For example, the more positive the query is, the more the user is engaged in the current application.
11. Is image uploaded during the usage of one application. In this case, it is likely to link an application with the image.
12. The number of users that current application is used by.
13. Is the application a third-party application or not. For example, the weight of a third-party application may be enhanced if the third-party application developer pays money for that. In some implementations, the third-party application developer may bid for special trigger words to help triggering their applications.

[0080] By using the features that may be abstracted from the user-application usage database, the similarity computing model may be implemented with machine learning algorithm. An example of the machine learning algorithm may be a gradient boosting decision tree algorithm used to compute the similarity scores.

[0081] In some implementations, when current user's application usage data is quite limited or even empty, the statistical information of all the existing users may be used as features for computing the similarity scores for application recommendation. In this case, the most popular applications which are used by most users or have relatively long usage time are likely to be recommended to the current user.

[0082] Figure 7 illustrates an exemplary collection of information from user data according to an embodiment. A user clustering model may be used to automatically cluster users into a number of groups and further link each group with a list of keywords which further is connected with events. The word "event" is to express a group of sentences or sentence-level descriptions that share a common topic key word.

[0083] A Latent Dirichlet allocation (LDA) clustering algorithm may be used at 704 to automatically cluster user data 702 into a plurality of clusters such as cluster 1 706, cluster 2 708 and so on. The user data 702 may be user's log data stored in the user log database 272 and/or user-application usage data stored in the user-application usage database 274. In some implementations, all users' log data (e.g., in the form of query-response pairs) and user's queries stored in the user-application stage database are collected as the user data 702.

[0084] In order to make the user data time-sensitive, a discounted weight may be assigned to older user data by a value of λ/n, where λ takes a value of (0, 1] and n is the time distance (e.g., number of months) of the user data till the time of clustering. For example, λ may be 0.8 and then for last month's logs of user data, the weight is 0.8 and for logs of user data two months ago, the weight is $0.8/2 = 0.4$. Through this way, ideas collected from the user clustering process may be "fresh".

[0085] Each cluster include a list of keywords, such as My Neighbor Totoro, One Piece, ..., shown at 706 and Spring roll, Curry, ..., shown at 708. It should be appreciated that although two clusters are shown in the Figure 7, there may be a certain number of clusters, for example the number of clusters may be in hundreds level, thousands level, tens of thousands level, or even millions level. A label may be explicitly defined for each cluster in some implementations. The clusters may just be annotated by integer numbers in some other implementations. Therefore the clustering algorithm may be carried on by only predefining the number of clusters without other configurations.

[0086] Each keyword in each cluster is appended with a list of one or more events.

An exemplary keyword "My Neighbor Totoro" in cluster 1 is shown at 710, the keyword "My Neighbor Totoro" is appended with two exemplary events "Rinna, I want to see the movie of "My Neighbor Totoro"" and "Rinna, can you find the movie "My Neighbor Totoro" for me" shown at 712 and 714. It should be appreciated that there may be more or less events appended to a keyword. In some implementations, user queries from the user data 702, which match a keyword of a cluster, may be appended to the keyword of the cluster as the events, and users that sent the queries may be linked to the cluster. Taking the keyword "My Neighbor Totoro" in cluster 1 as an example, a query "Rinna, I want to see the movie of My Neighbor Totoro" contains the keyword "My Neighbor Totoro", then the query is appended to the keyword as the event and accordingly the query as well as users that sent the query is linked with the cluster 1. All the users that send queries containing the keyword of "My Neighbor Totoro" may be included in the cluster 1. Similarly, all the users that send queries containing the keyword of "One Piece" may be included in the cluster 1. Finally all the users that send queries containing any keyword of cluster 1 may be included in the cluster 1.

[0087] The user clustering model is designed to better understand the frequently used words and frequently attended events by large-scale users. And the information collected by the user clustering is helpful for application developers to develop new applications that fit for the habits of each group of users, so as to improve the engagement rate of the new developed applications.

[0088] Figure 8 illustrates an exemplary collection of information from web data according to an embodiment.

[0089] A LDA clustering model may be used at 804 to automatically cluster web data 802 into a plurality of clusters such as cluster 1 806, cluster 2 808 and so on. The web data 802 may be from various web resources, examples of which may include social networks, knowledge-related website and so on.

[0090] Each cluster include a list of keywords, such as Star War, My Neighbor Totoro, ..., shown at 806 and Spring roll, sushi, …, shown at 808. It should be appreciated that although two clusters are shown in the Figure, there may be a certain number of clusters, as discussed above.

[0091] Each keyword in each cluster is appended with a list of one or more events. An exemplary keyword "My Neighbor Totoro" in cluster 1 is shown at 810, the

keyword "My Neighbor Totoro" is appended with two exemplary events ""My Neighbor Totoro" is a long animation movie produced by Studio Ghibli." and ""My Neighbor Totoro" is directed by Hayao Miyazaki." shown at 812 and 814. It should be appreciated that there may be more or less events appended to a keyword. In some implementations, sentence level descriptions (such as sentences 812, 814 or sentences 818, 820) from the web data 802, which match a keyword of a cluster, may be appended to the keyword of the cluster as the events. As illustrated, the sentences 812 and 814 are appended to the keyword 810 in the cluster 1, and the sentences 818 and 820 are appended to the keyword 816 in the cluster 2.

[0092] Figure 9 illustrates an exemplary link of user clusters and web cluster according to an embodiment. The same label numbers shown in Figures 7-9 denote same elements.

[0093] After generating a list of user clusters from user data as shown in Figure 7 and a list of web clusters from web data as show in Figure 8, similar clusters in these two lists of clusters may be unified. For example, as shown in Figure 9, if a keyword (e.g., "My Neighbor Totoro" 710) is shared by a cluster in the user cluster list (e.g., cluster 1 in the user cluster list) and a cluster in the web cluster list (e.g., cluster 1 in the web cluster list), then the events (e.g., events 812, 814) appended to the keyword in the web cluster may be linked to the events appended to the keyword in the user cluster. This linking is meaningful, since user's intention of requirement is mostly included in the events from the user and the events from the web may supply materials for preparing user's special intentions, which is identified by the keywords. Accordingly, the disclosure builds new query-answer pairs by making use of the link of the user data and web data, where a query is from users and an answer is from the web.

[0094] In some implements, a query-answer similarity score computing model is used to find high confidence answers from web data for queries from user data. An exemplary algorithm of the model is as follows.

For each user cluster Cu in user cluster list:

Find web cluster Cw in web cluster list so that Cw shares the maximum number of words with Cu;

For each query Qu that is connected with Cu:

$sim_{max}(Qu, Sw) = 0$;

For each sentence Sw that is connected with Cw:

Compute the similarity score sim(Qu, Sw) and record the Sw if sim(Qu, Sw) > $sim_{max}(Qu, Sw)$;

$sim_{max}(Qu, Sw) = sim(Qu, Sw)$;

If $sim_{max}(Qu, Sw)$ >= threshold

Take (Qu, Sw) as a <query, answer> candidate pair.

[0095] The sim(Qu, Sw) denotes the similarity score between Qu and Sw, $sim_{max}(Qu, Sw)$ denotes the maximum similarity score. A machine learning module, for example, a gradient boosting decision tree, may be trained for computing the similarity score sim(Qu, Sw), which indicates how good the answer sentence responds to a query. The query Qu is an event in the user cluster Cu, and the sentence Sw is an event in the web cluster Cw. The threshold is a predefined value, an example of the threshold may be 0.5, where the sim(Qu, Sw) ranges from 0 to 1.

[0096] It should be appreciated that the above algorithm is illustrative without limitation to the scope of the disclosure and suitable modification may be made to the algorithm. For example, one best sentence Sw from the web cluster may be found for one query from the user cluster by using the above algorithm, however as a variation of the algorithm, the maximum similarity score $sim_{max}(Qu, Sw)$ is not considered, and all sentences from the web cluster having scores larger than the threshold may be taken as candidate answers for the query, therefore it's possible that multiple sentences may be found from the web cluster as candidate answers for one query from the user cluster. It should be appreciated that the finding of candidate query-answer pairs may be performed periodically based on user data and web data.

[0097] As discussed above, the number of clusters may be in hundreds level, thousands level, tens of thousands level, or even millions level, therefore rich candidate query-answer candidate pairs may be found through the above process by using the link of the user data and web data. In some implementations, the candidate query-answer pairs may be used to update applications.

[0098] Figure 10 illustrates an exemplary process for updating an application according to an embodiment.

[0099] At 1010, the chatbot identifies a user query is matched with an application, similarly as the YES decision made at 506 of Figure 5.

[00100] At 1020, the matched application is activated, similarly as the process at 520 of Figure 5.

[00101] At 1030, the chatbot identifies whether there are sufficient available responses for the application. In some implementations, a decision of insufficient available responses may be made if the number of the available responses for the application is smaller than a threshold. For example, for a movie topic related application, it may identify that the available responses are insufficient when the number of the available responses is smaller than the threshold, and it may further determine that the application may be updated by complementing movie data from web.

[00102] At 1040, the query may be linked to a web cluster. The link of the query and the web cluster may be implemented based on the keywords or events of the clusters. Taking the query "Rinna, can you find the movie "My Neighbor Totoro" for me" as an example, this query may be used to trigger a movie-related application at 1020, and may be linked to a web cluster related to movie topics.

[00103] At 1050, data may be obtained from the web cluster to update the application such as the movie-related application. In some implementations, the candidate query-answer pairs discussed above corresponding to the sentences of the web cluster may be obtained. As discussed above, since user's intention of requirement is mostly included in the events from the user and the events from the web may supply materials for preparing user's special intentions, the candidate query-answer pairs obtained for the application may be used to prepare user's various intentions with web materials. In some other implementations, the sentences in the web cluster which are not included in the available responses of the application may be obtained. Taking the movie-related application as an example, data about new movies from the web may be obtained to update the application.

[00104] At 1060, the data obtained from the web cluster are appended to the application.

[00105] Although the updating of an application is performed as needed after the application is triggered as illustrated in Figure 10, it may be possible that the updating of an application may be performed periodically based on the candidate query-answer

pairs, which are also updated periodically.

[00106] It should be appreciated that reasonable modification may be made to the process of updating an application. For example, after the matched application is activated at 1020, if a second user query is received but a response cannot be obtained from the application's data, it may determine that the available responses is insufficient at 1030, then a response for the second user query may be obtained from web data or web cluster, and meanwhile the application may be updated by collecting data from the web data, as discussed at 1040 to 1060.

[00107] In addition to the update of existing application by using the user clusters and the web clusters which are obtained from user data and web data, the user clusters and web clusters may also be used to drive idea collection about applications. Specifically, since web data is updating in a fast speed and the chatbot is not guaranteed to include the novel events or hot keywords such as new named entities, new games, new movies and so on, the web cluster list may be used to collect novel ideas for constructing new applications for the chatbot. For example, for one web cluster, there are quite a few keywords appearing in users' log data and/or there is not a corresponding cluster from user cluster list, then the event sentences from the web may be used as materials or data resources for preparing new applications.

[00108] In addition, the user clustering may also be used to improve the application recommendation performed by the application recommendation module 280 since the user clustering is helpful for better understanding the frequently used words and frequently attended events by large-scale users. In addition to the above mentioned features obtained from the user-application usage database, one or more of the following features may be used in the machine learning module implemented at the application recommendation module 280 for computing similarity scores between a query and trigger data of applications:

1. The clusters that current user belongs to.
2. The type of an application. Examples of the application type may comprise a time-sensitive application, game-style application, functional application such as weather report application, image-based application, and other suitable types of application.
3. The edit distance in word level between a query and the trigger word list.
4. The edit distance in character level between a query and a trigger word;

5. The word2vec similarity score between a query and a trigger word.
6. BM25 score between a query and a trigger word.

[00109] Figure 11 illustrate an exemplary process for updating key words of an application according to an embodiment.

[00110] At 1110, the chatbot identifies a user query is matched with an application.

[00111] At 1120, the chatbot recommends the application to the user. The recommendation may be implemented as the process at 516 of Figure 5 when there is a current active application. The recommendation may also be performed whenever the matching decision at 1110 is made, irrespective whether there is a current active application or not.

[00112] At 1130, the chatbot receives user's feedback in response to the recommendation of the application.

[00113] At 1140, the chatbot may identify whether the user's feedback is positive or negative, for example, by using a sentiment analysis module.

[00114] At 1150, the matched application is activated in response to a positive feedback.

[00115] At 1160, the chatbot may identify whether the application is ended normally. In some implementations, the application is identified to be ended normally when the application is finished without interruption and/or the usage time of the application is above average usage time. The average usage time may be average usage time of the application among multiple users, and may also be average usage time of the application by the current user.

[00116] At 1170, the user query is added to the keyword list of the application as a new keyword of the application. Accordingly the keywords of the application may self-grow along with the usage of the application.

[00117] It should be appreciated that although the term keyword is used in the description, the keywords of an application are not limited to a word-level keyword, actually the keywords of an application may be phase-level keywords, sentence-level keywords, or pattern-level keywords, which may be collectively referred to as trigger data of the application.

[00118] It should be appreciated that reasonable modification may be made to the process of updating key words. For example, the user query may be added to the

keyword list of the application at 1170 when determining a positive feedback to the recommendation is received at 1150, without considering whether the application is ended normally at 1160.

[00119] Figure 12 illustrates an exemplary process for application recommendation through intelligent automated chatting according to an embodiment.

[00120] At 1210, a query is received in a conversation. At 1220, matching rates between the query and applications are scored. At 1230, the query is identified to be matched with a first application of the applications based on the scores. At 1240, the first application is recommended in the conversation.

[00121] In some implementations, the query may be added into trigger data of the first application in response to a positive answer to the recommendation. In some other implementations, the query may be added into the trigger data of the first application if the first application is finished without interruption and/or the usage time of the first application is above an average usage time.

[00122] In some implementations, the recommending the first application may be implemented by outputting a question about whether to activate the first application. In some implementations, the recommending the first application may be implemented by activating the first application.

[00123] In some implementations, it may identify that a second application is active after identifying the query is matched with the first application. The recommending the first application may be implemented by outputting a question about whether to switch from the second application to the first application. A switch from the second application to the first application may be performed in response to a positive answer to the question. The second application may be continued in response to a negative answer to the question.

[00124] In some implementations, usage data of the first application and/or the second application associated with the user may be recorded in a user-application usage database.

[00125] In some implementations, a second query is received while the first application is active. A second response to the second query may be obtained from web data and presented to the user. The second query and the second response may be appended to the first application, so as to update the first application's query-response pairs.

[00126] In some implementations, it identifies that available query-response pairs for the first application are insufficient. Candidate query-response pairs are appended to the first application, so as to update the first application's query-response pairs. Queries of the candidate query-response pairs are from user data and responses of the pairs are from a web data.

[00127] In some implementations, the query is determined to be matched with the first application based on a matching rate between the query and the first application and a threshold of the first application. The threshold of the first application may be updated based on the number of successful usages of the first application and the number of unsuccessful usages of the first application.

[00128] In some implementations, the matching rates between the query and the applications are scored based on the query, trigger data of the applications, and at least one of user-application usage data of the applications, user profile, monetary bid information of the applications, types of the applications, latent semantic scores between the query and the trigger data. The user profile comprises user cluster information indicating which of a plurality of clusters the user belongs to, wherein each cluster includes a list of users and a list of keywords.

[00129] Figure 13 illustrates an exemplary process for information collection according to an embodiment.

[00130] At 1310, a plurality of user clusters are generated based on user data associated with a service, wherein a user cluster comprises a set of users and a set of key words, and a key word in the user cluster is appended with one or more sentence-level descriptions. At 1320, the plurality of user clusters are stored in a database.

[00131] An example of the service may be a chatbot service or AI chatting service as discussed above. It may be appreciated that the information collection process may be applicable to chatting service such IM service, through which users chat with each other rather than chat with a chatbot. The user clusters obtained from the user data of such chatting service may also be used to reflect users' intention of requirement.

[00132] In some implementations, the user data may be clustered into a plurality of user clusters. For a key word in a user cluster, one or more sentence-level descriptions from one or more users in the user data are appended to the key word of the user cluster, and the one or more users are included in the user cluster.

[00133] In some implementations, a plurality of web clusters are generated based

on web data. A web cluster includes a list of key words, and a key word in the web cluster is appended with one or more sentence-level descriptions.

[00134] In some implementations, the web data may be clustered into a plurality of web clusters. For a key word in a web cluster, one or more sentence-level descriptions from one or more users in the web data are appended to the key word of the web cluster.

[00135] In some implementations, for a key word which is included in both a user cluster and a web cluster, the sentence-level descriptions of the key word of the web cluster are appended to the key word of the user cluster.

[00136] In some implementations, query-response pairs are obtained based on the plurality of user clusters and the plurality of web clusters. Queries of the pairs are from the user clusters and responses of the pairs are from the web clusters. Applications associated with the service may be updated by using the query-response pairs.

[00137] In some implementations, the user data may comprise at least one of user chatting log data and user-application usage data. Information indicating recommendation of application development may be generated based on at least one of the user data (i.e., the historical user chatting log data and user-application usage data) and web data. For example, one or more web clusters which contain a number of keywords appearing in the user clusters are identified. One or more web clusters which do not correspond to any of the user clusters are identified. Therefore the information collection from at least one of the user data and the web data facilitates the application recommendation in one aspect and provides information for application development in another aspect.

[00138] Figure 14 illustrates an exemplary apparatus for application recommendation through intelligent automated chatting according to an embodiment.

[00139] The apparatus 1400 comprises an obtaining module 1410 and a recommending module 1420. The obtaining module 1410 may obtain a query in a conversation. The recommending module 1420 may score matching rates between the query and applications, identify that the query is matched with a first application of the applications based on the matching rates, and recommend the first application in the conversation.

[00140] In some implementations, the recommending module 1420 may

recommend the first application by outputting a question about whether to activate the first application. In some implementations, the recommending module 1420 may recommend the first application by activating the first application.

[00141] In some implementations, the recommending module 1420 may identify a second application is active after identifying the query is matched with the first application. The recommending module 1420 may recommend the first application by outputting a question about whether to switch from the second application to the first application. The recommending module1420 may switch from the second application to the first application in response to a positive answer to the recommendation. The recommending module 1420 may continue the second application in response to a negative answer to the recommendation.

[00142] It should be appreciated that the recommending module 1420 may perform any operations related to application recommendation according to the various embodiments as mentioned above in connection with Figures 1-13. It should be appreciated that the apparatus 1400 may also comprise any other modules configured for performing any operations of the methods for application recommendation according to the various embodiments as mentioned above in connection with Figures 1-13.

[00143] Figure 15 illustrates an exemplary apparatus for information collection according to an embodiment.

[00144] The apparatus 1500 comprises a generating module 1510 and a storing module 1520. The generating module 1510 may generating a plurality of user clusters based on user data associated with a service. A user cluster comprises a set of users and a set of key words, and a key word in the user cluster is appended with one or more sentence-level descriptions. The storing module 1520 may store the plurality of user clusters in a database.

[00145] It should be appreciated that the generating module 1510 may perform any operations related to information collection according to the various embodiments as mentioned above in connection with Figures 1-13. It should be appreciated that the apparatus 1500 may also comprise any other modules configured for performing any operations of the methods for information collection according to the various embodiments as mentioned above in connection with Figures 1-13.

[00146] Figure 16 illustrates an exemplary computing system according to an

embodiment.

[00147] The system 1600 may comprise one or more processors 1610. The system 1600 may further comprise a memory 1620 that is connected with the one or more processors 1610.

[00148] The memory 1620 may store computer-executable instructions that, when executed, cause the one or more processors 1610 to receive a query in a conversation, score matching rates between the query and applications, identify that the query is matched with a first application of the applications based on the matching rates, and recommend the first application in the conversation.

[00149] The memory 1620 may store computer-executable instructions that, when executed, cause the one or more processors 1610 to generate a plurality of user clusters based on user data associated with a service and store the plurality of user clusters in a database, wherein a user cluster comprises a set of users and a set of key words, and a key word in the user cluster is appended with one or more sentence-level descriptions.

[00150] It should be appreciated that the computer-executable instructions, when executed, cause the one or more processors 1610 to perform any operations of the processes according to the embodiments as mentioned above in connection with Figures 1-13.

[00151] The embodiments of the present disclosure may be embodied in a non-transitory computer-readable medium. The non-transitory computer-readable medium may comprise instructions that, when executed, cause one or more processors to perform any operations of the processes according to the embodiments as mentioned above.

[00152] It should be appreciated that all the operations in the processes described above are merely exemplary, and the present disclosure is not limited to any operations in the processes or sequence orders of these operations, and should cover all other equivalents under the same or similar concepts.

[00153] It should also be appreciated that all the modules in the apparatuses described above may be implemented in various approaches. These modules may be implemented as hardware, software, or a combination thereof. Moreover, any of these modules may be further functionally divided into sub-modules or combined together.

[00154] Processors have been described in connection with various apparatuses

and methods. These processors may be implemented using electronic hardware, computer software, or any combination thereof. Whether such processors are implemented as hardware or software will depend upon the particular application and overall design constraints imposed on the system. By way of example, a processor, any portion of a processor, or any combination of processors presented in the present disclosure may be implemented with a microprocessor, microcontroller, digital signal processor (DSP), a field-programmable gate array (FPGA), a programmable logic device (PLD), a state machine, gated logic, discrete hardware circuits, and other suitable processing components configured to perform the various functions described throughout the disclosure. The functionality of a processor, any portion of a processor, or any combination of processors presented in the present disclosure may be implemented with software being executed by a microprocessor, microcontroller, DSP, or other suitable platform.

[00155] Software shall be construed broadly to mean instructions, instruction sets, code, code segments, program code, programs, subprograms, software modules, applications, software applications, software packages, routines, subroutines, objects, threads of execution, procedures, functions, etc. The software may reside on a computer-readable medium. A computer-readable medium may include, by way of example, memory such as a magnetic storage device (e.g., hard disk, floppy disk, magnetic strip), an optical disk, a smart card, a flash memory device, random access memory (RAM), read only memory (ROM), programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), a register, or a removable disk. Although memory is shown separate from the processors in the various aspects presented throughout the present disclosure, the memory may be internal to the processors (e.g., cache or register).

[00156] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein. All structural and functional equivalents to the elements of the various aspects described throughout the present disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims.

WHAT IS CLAIMED IS:

1. A method for application recommendation through intelligent automated chatting, comprising:

receiving a query in a conversation;

scoring matching rates between the query and applications;

identifying that the query is matched with a first application of the applications based on the matching rates; and

recommending the first application in the conversation.

2. The method of claim 1, the recommending the first application further comprises performing one of the following:

outputting a question about whether to activate the first application; or

activating the first application.

3. The method of claim 1, further comprising:

identifying a second application is active after identifying the query is matched with the first application;

the recommending the first application further comprises outputting a question about whether to switch from the second application to the first application; and

the method further comprises performing one of the following:

switching from the second application to the first application in response to a positive answer to the question, or

continuing the second application in response to a negative answer to the question.

4. The method of claim 3, further comprising:

recording usage data of the first application and/or the second application associated with the user in a user-application usage database.

5. The method of claim 3, further comprising:

receiving a second query while the first application is active;

obtaining a second response to the second query from web data;

outputting the second response; and

appending the second query and the second response to the first application.

6. The method of claim 1, further comprising performing one of the following:

adding the query into trigger data of the first application in response to a positive answer to the recommendation; or

adding the query into the trigger data of the first application if the first application is finished without interruption and/or the usage time of the first application is above an average usage time.

7. The method of claim 1, further comprising:

identifying that available query-response pairs for the first application are insufficient; and

appending candidate query-response pairs to the first application, where queries of the candidate query-response pairs are from user data and responses of the pairs are from a web data.

8. The method of claim 1, the identifying the query is matched with the first application further comprises:

identifying the query is matched with the first application based on a matching rate between the query and the first application and a threshold of the first application.

9. The method of claim 8, wherein the threshold of the first application is updated based on the number of successful usages of the first application and the number of unsuccessful usages of the first application.

10. The method of claim 1, wherein the scoring further comprises:

scoring the matching rates based on the query, trigger data of the applications, and at least one of user-application usage data of the applications, user profile, monetary bid information of the applications, types of the applications, latent semantic scores between the query and the trigger data.

11. The method of claim 10, wherein the user profile comprises user cluster information indicating which of a plurality of clusters the user belongs to.

12. A method for collecting information, comprising:

generating a plurality of user clusters based on user data associated with a service, wherein a user cluster comprises a set of users and a set of key words, and a key word in the user cluster is appended with one or more sentence-level descriptions; and

storing the plurality of user clusters in a database.

13. The method of claim 12, further comprising:

generating a plurality of web clusters based on web data, wherein a web cluster includes a list of key words, and a key word in the web cluster is appended with one or more sentence-level descriptions.

14. The method of claim 13, further comprising

obtaining query-response pairs based on the plurality of user clusters and the plurality of web clusters, wherein queries of the pairs are from the user clusters and responses of the pairs are from the web clusters.

15. The method of claim 14, further comprising

updating applications associated with the service by using the query-response pairs.

16. The method of claim 13, wherein the user data comprises user chatting log data and user-application usage data, and the method further comprises:

generating information indicating recommendation of application development based on at least one of the user clusters and the web clusters.

17. The method of claim 16, wherein the generating information further comprises:

identifying one or more web clusters which contains a number of keywords appearing in the user clusters; and/or

identifying one or more web clusters which do not correspond to any of the user clusters.

18. An apparatus for application recommendation through intelligent automated chatting, comprising:

an obtaining module for obtaining a query in a conversation; and

a recommending module for scoring matching rates between the query and applications, identifying that the query is matched with a first application of the applications based on the matching rates, and recommending the first application in the conversation.

19. The apparatus of claim 18, wherein recommending module is further for recommending the first application by performing one of the following:

outputting a question about whether to activate the first application; or

activating the first application.

20. The apparatus of claim 18, wherein recommending module is further for:

identifying a second application is active after identifying the query is matched with the first application;

recommending the first application by outputting a question about whether to switch from the second application to the first application; and

performing one of the following:

switching from the second application to the first application in response to a positive answer to the question, or

continuing the second application in response to a negative answer to the question.

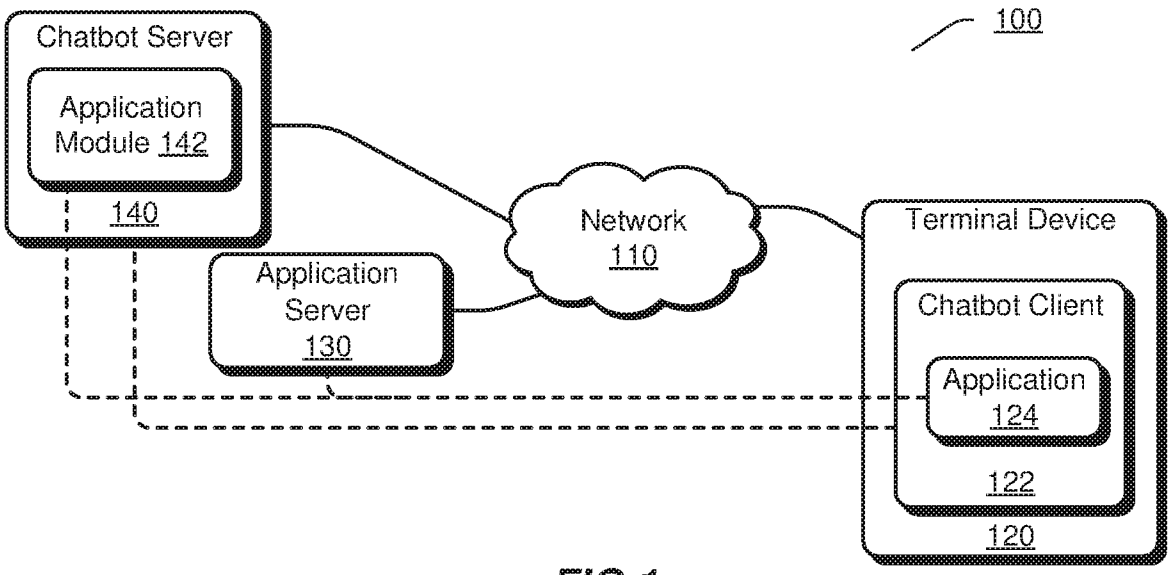
100
Chatbot Server
Application Module 142
140
Network 110
Application Server 130
Terminal Device
Chatbot Client
Application 124
122
120

FIG 1

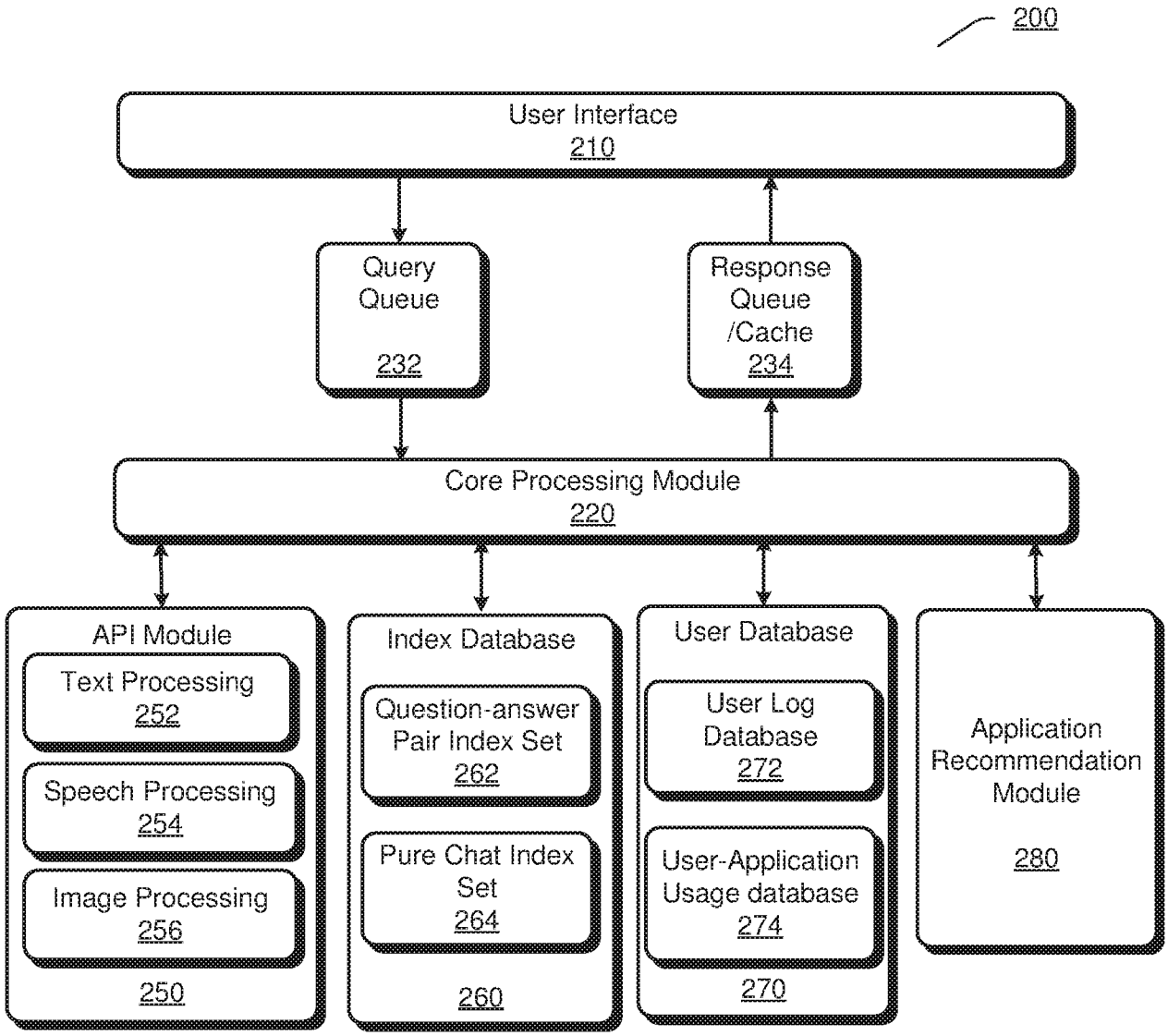
200
User Interface 210
Query Queue 232
Response Queue /Cache 234
Core Processing Module 220
API Module
Text Processing 252
Speech Processing 254
Image Processing 256
250
Index Database
Question-answer Pair Index Set 262
Pure Chat Index Set 264
260
User Database
User Log Database 272
User-Application Usage database 274
270
Application Recommendation Module 280

FIG 2

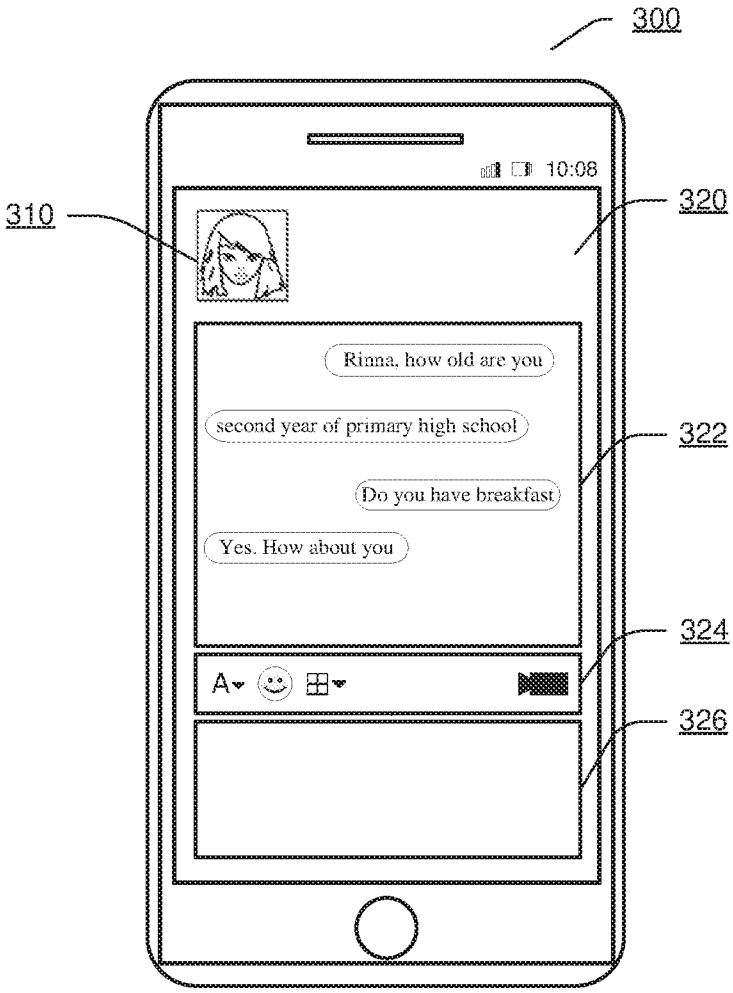
300
10:08
310
320
Rinna, how old are you
second year of primary high school
322
Do you have breakfast
Yes. How about you
324
326

FIG 3

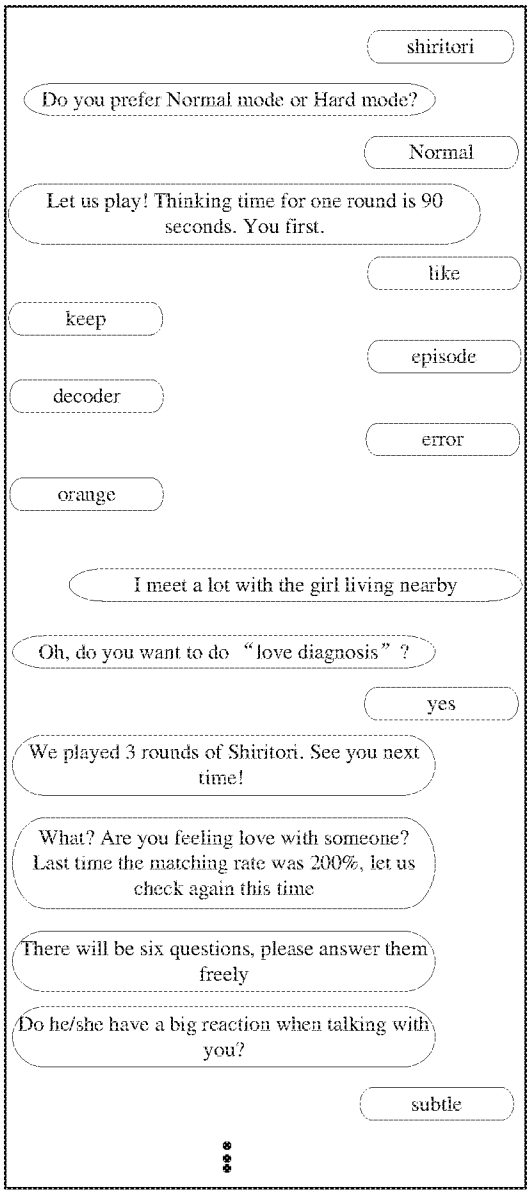
shiritori
Do you prefer Normal mode or Hard mode?
Normal
Let us play! Thinking time for one round is 90 seconds. You first.
like
keep
episode
decoder
error
orange
I meet a lot with the girl living nearby
Oh, do you want to do "love diagnosis"?
yes
We played 3 rounds of Shiritori. See you next time!
What? Are you feeling love with someone? Last time the matching rate was 200%, let us check again this time
There will be six questions, please answer them freely
Do he/she have a big reaction when talking with you?
subtle

FIG 4

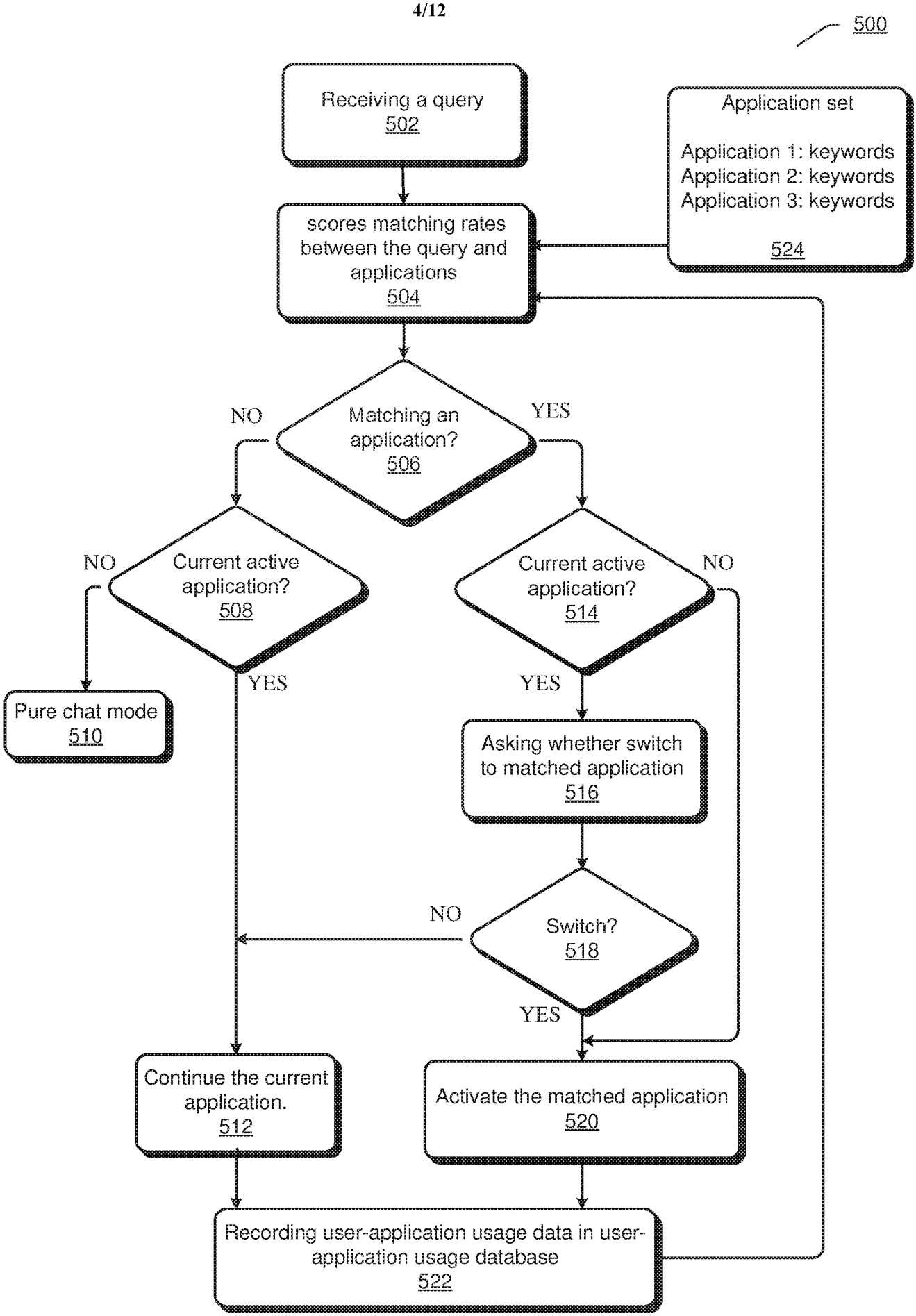
500
Receiving a query
502
Application set
Application 1: keywords
Application 2: keywords
Application 3: keywords
524
scores matching rates between the query and applications
504
Matching an application?
506
NO
YES
Current active application?
508
NO
YES
Pure chat mode
510
Current active application?
514
NO
YES
Asking whether switch to matched application
516
Switch?
518
NO
YES
Continue the current application.
512
Activate the matched application
520
Recording user-application usage data in user-application usage database
522

FIG 5

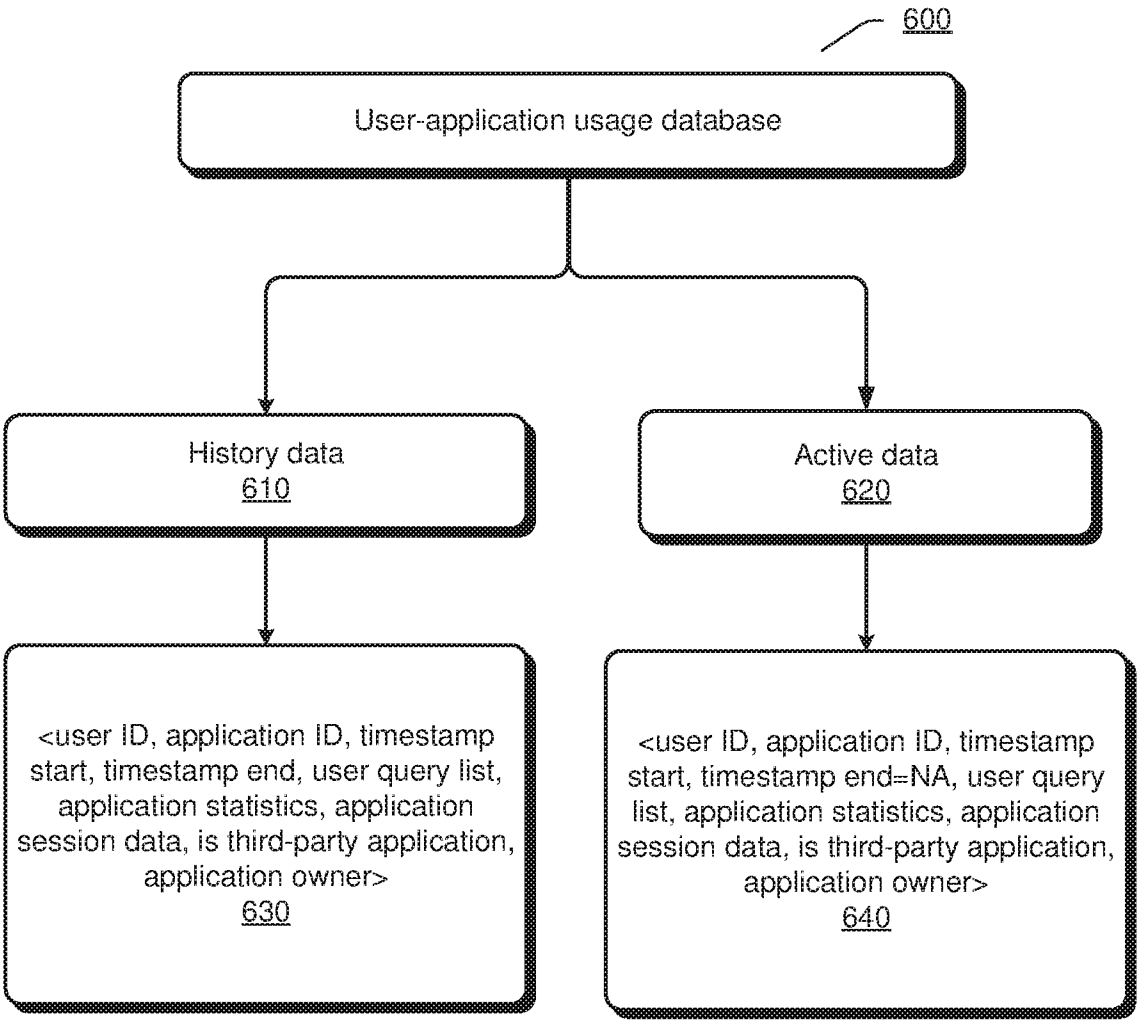
600
User-application usage database
History data
610
Active data
620
<user ID, application ID, timestamp start, timestamp end, user query list, application statistics, application session data, is third-party application, application owner>
630
<user ID, application ID, timestamp start, timestamp end=NA, user query list, application statistics, application session data, is third-party application, application owner>
640

FIG 6

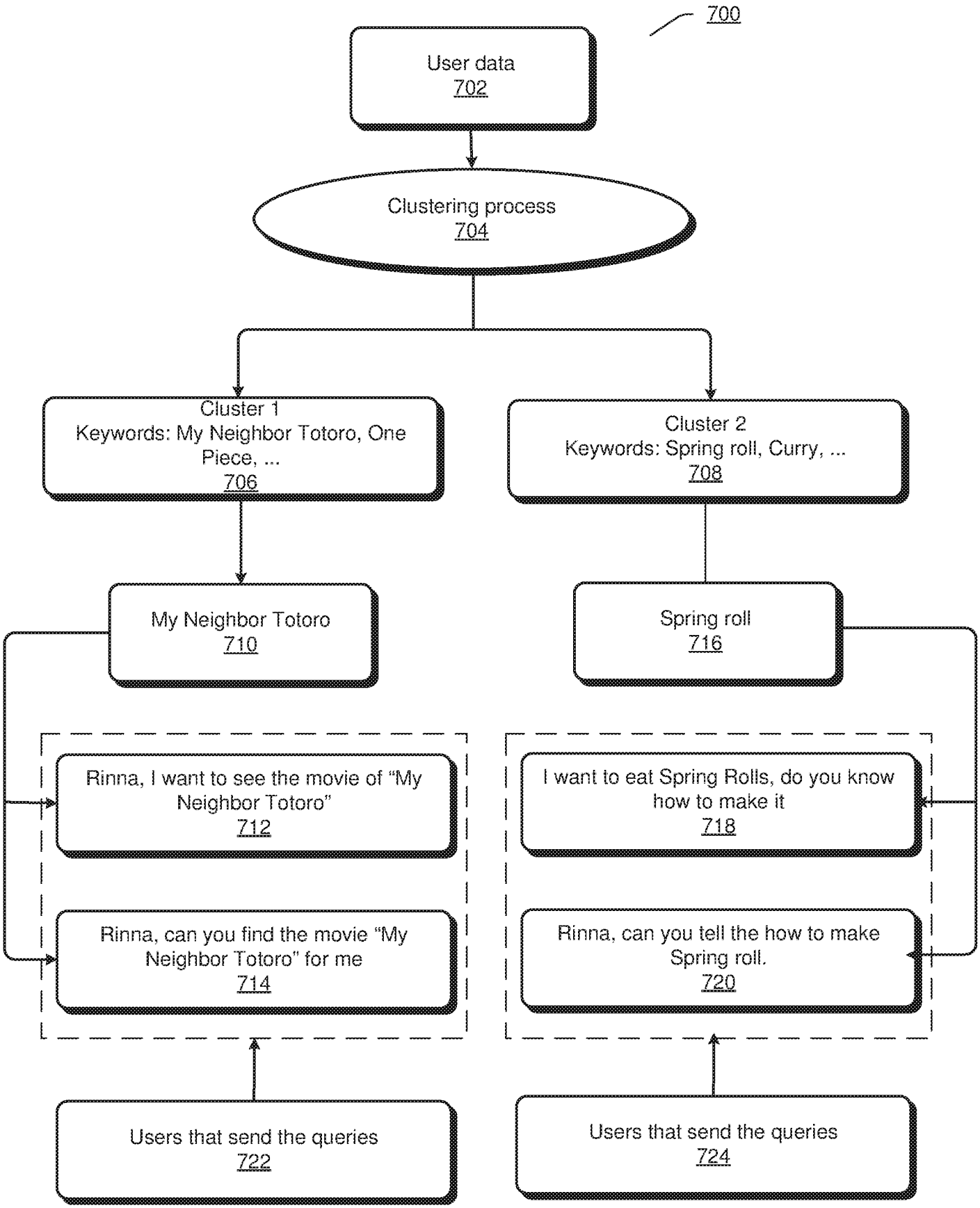
700
User data
702
Clustering process
704
Cluster 1
Keywords: My Neighbor Totoro, One Piece, ...
706
Cluster 2
Keywords: Spring roll, Curry, ...
708
My Neighbor Totoro
710
Spring roll
716
Rinna, I want to see the movie of "My Neighbor Totoro"
712
I want to eat Spring Rolls, do you know how to make it
718
Rinna, can you find the movie "My Neighbor Totoro" for me
714
Rinna, can you tell the how to make Spring roll.
720
Users that send the queries
722
Users that send the queries
724

FIG 7

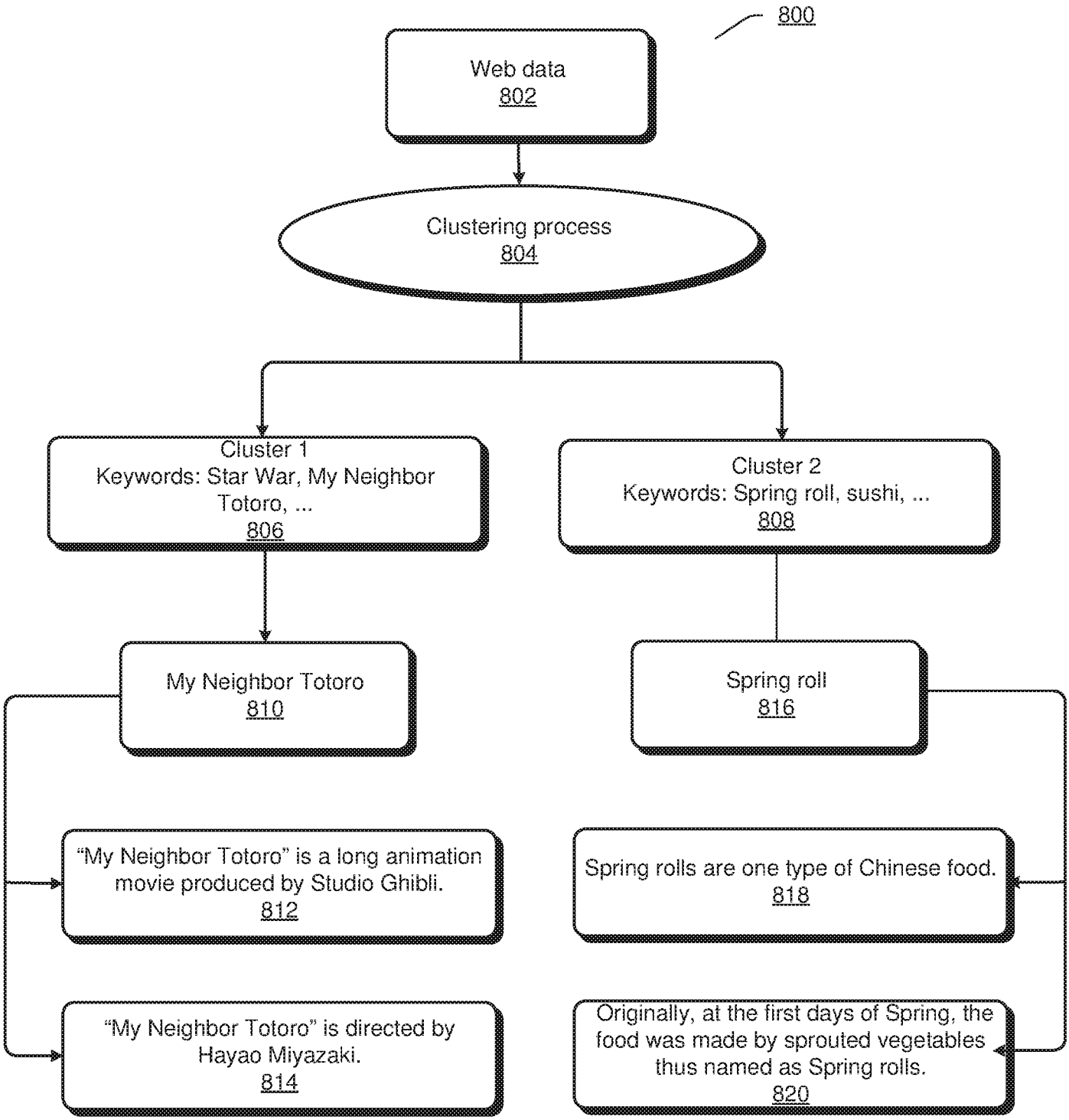
800
Web data
802
Clustering process
804
Cluster 1
Keywords: Star War, My Neighbor Totoro, ...
806
Cluster 2
Keywords: Spring roll, sushi, ...
808
My Neighbor Totoro
810
Spring roll
816
"My Neighbor Totoro" is a long animation movie produced by Studio Ghibli.
812
Spring rolls are one type of Chinese food.
818
"My Neighbor Totoro" is directed by Hayao Miyazaki.
814
Originally, at the first days of Spring, the food was made by sprouted vegetables thus named as Spring rolls.
820

FIG 8

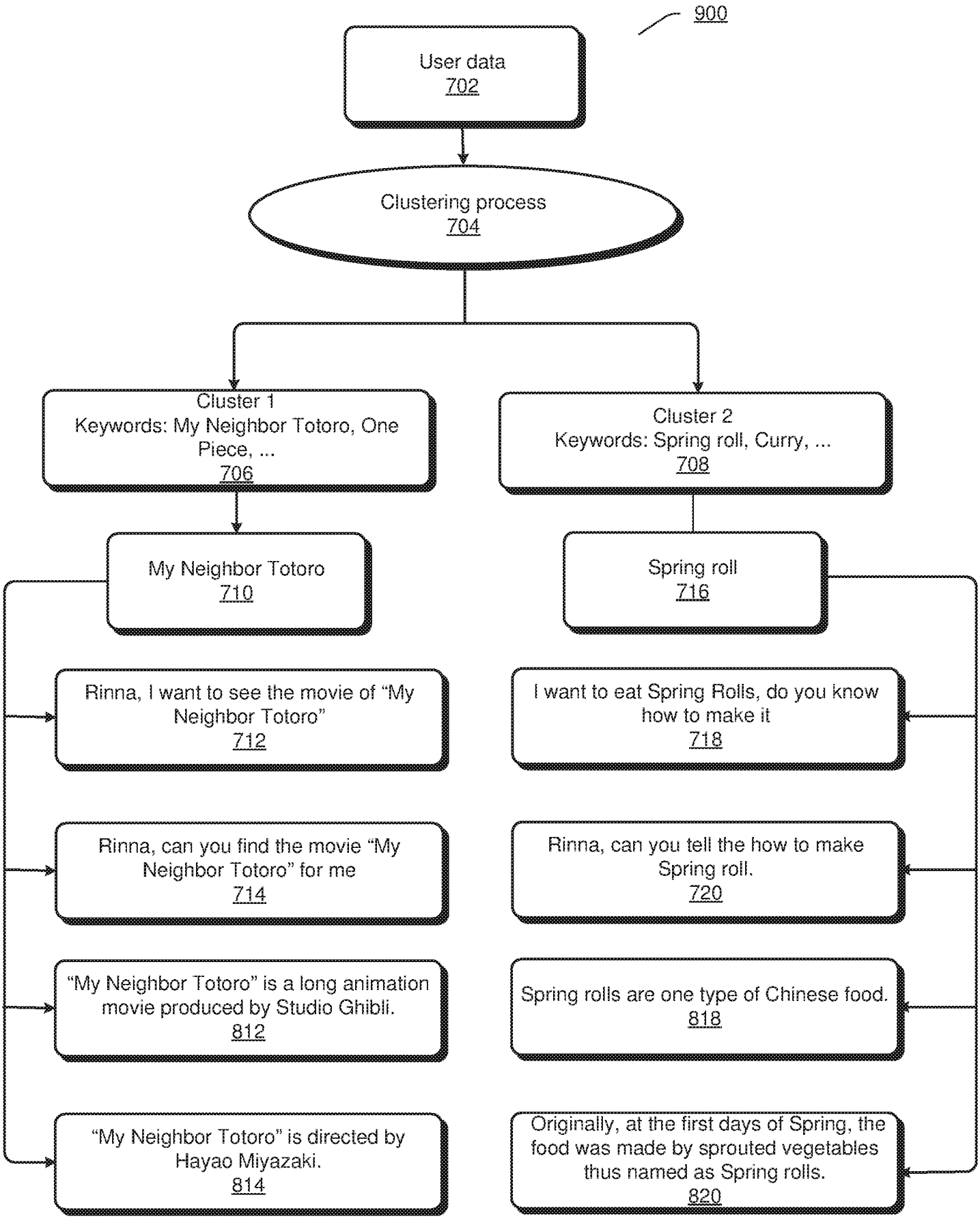
900
User data
702
Clustering process
704
Cluster 1
Keywords: My Neighbor Totoro, One Piece, ...
706
Cluster 2
Keywords: Spring roll, Curry, ...
708
My Neighbor Totoro
710
Spring roll
716
Rinna, I want to see the movie of "My Neighbor Totoro"
712
I want to eat Spring Rolls, do you know how to make it
718
Rinna, can you find the movie "My Neighbor Totoro" for me
714
Rinna, can you tell the how to make Spring roll.
720
"My Neighbor Totoro" is a long animation movie produced by Studio Ghibli.
812
Spring rolls are one type of Chinese food.
818
"My Neighbor Totoro" is directed by Hayao Miyazaki.
814
Originally, at the first days of Spring, the food was made by sprouted vegetables thus named as Spring rolls.
820

FIG 9

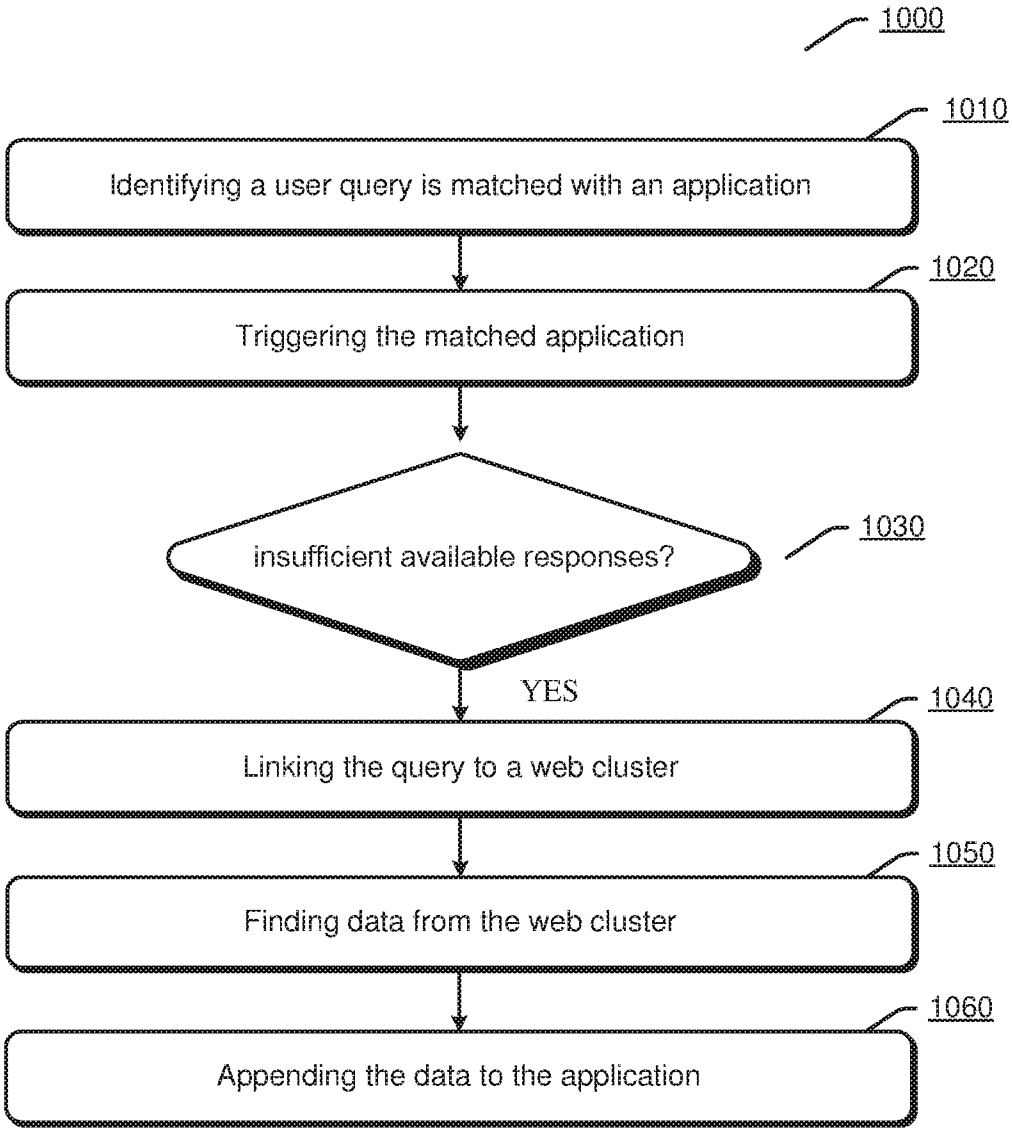
1000
1010
Identifying a user query is matched with an application
1020
Triggering the matched application
insufficient available responses?
1030
YES
1040
Linking the query to a web cluster
1050
Finding data from the web cluster
1060
Appending the data to the application

FIG 10

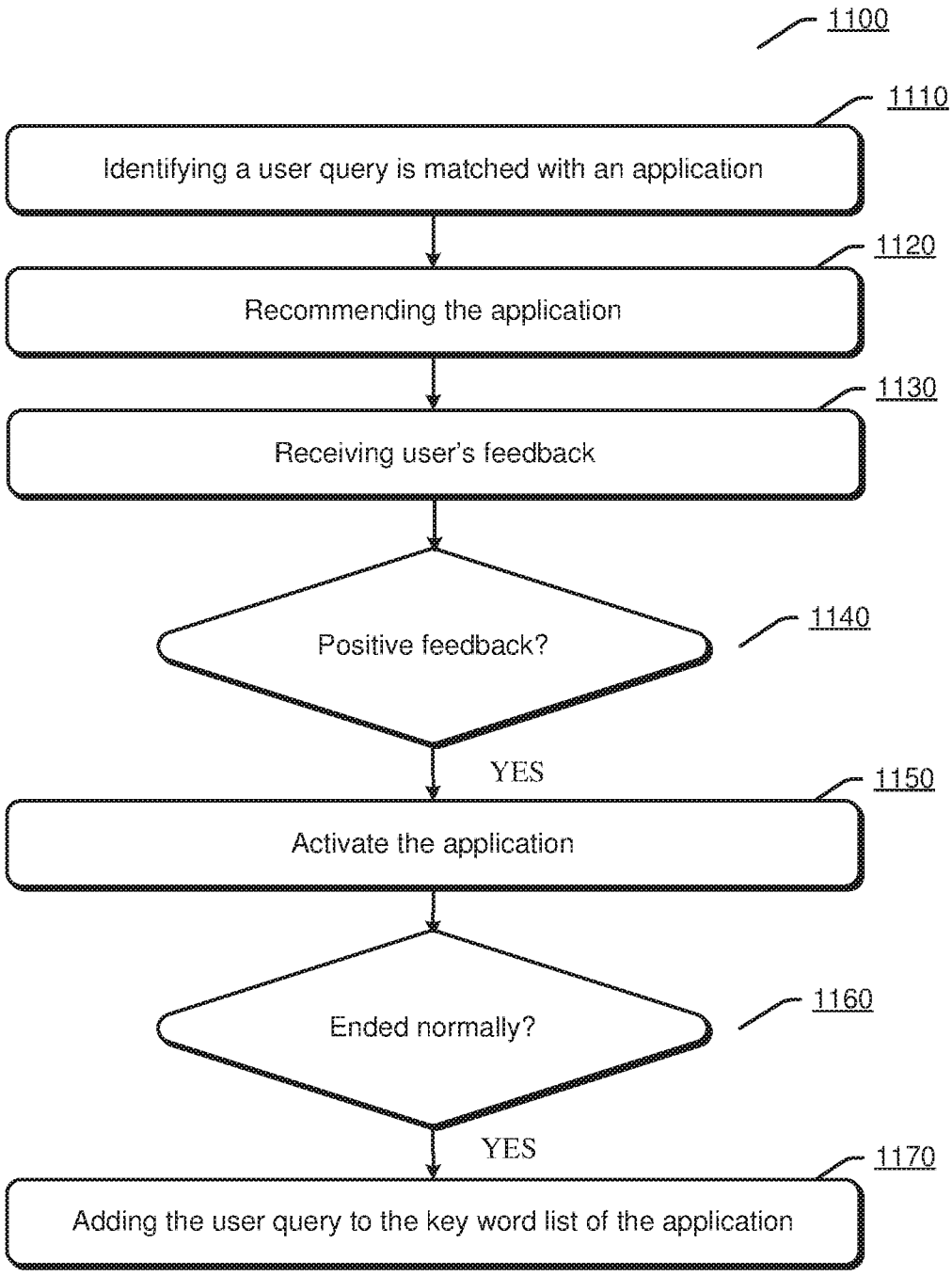
1100
1110
Identifying a user query is matched with an application
1120
Recommending the application
1130
Receiving user's feedback
Positive feedback?
1140
YES
1150
Activate the application
Ended normally?
1160
YES
1170
Adding the user query to the key word list of the application

FIG 11

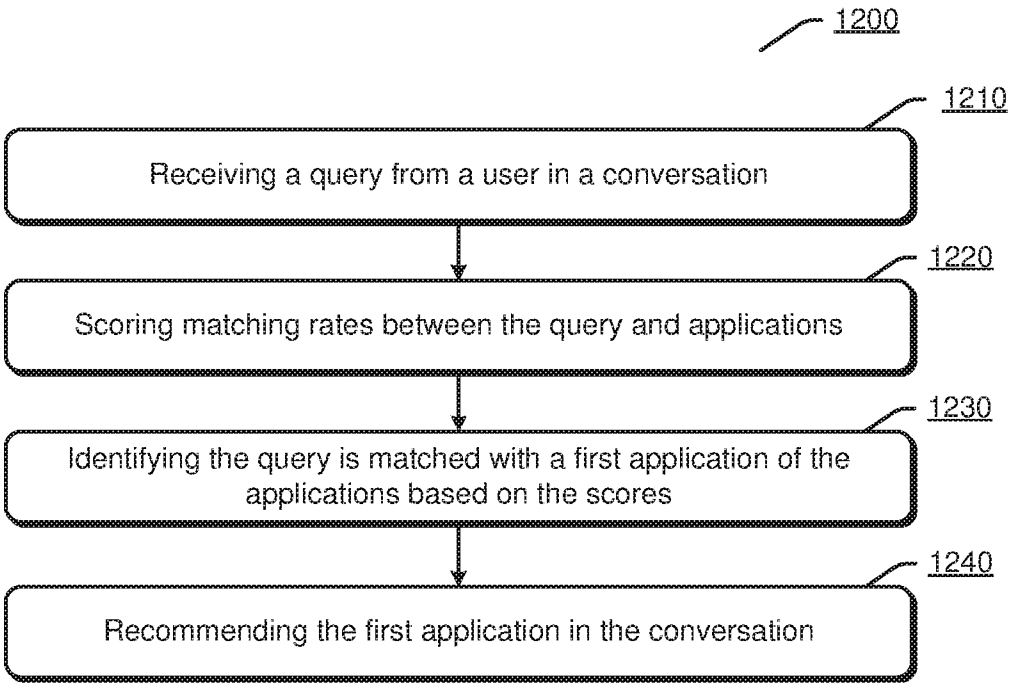
1200
1210
Receiving a query from a user in a conversation
1220
Scoring matching rates between the query and applications
1230
Identifying the query is matched with a first application of the applications based on the scores
1240
Recommending the first application in the conversation

FIG 12

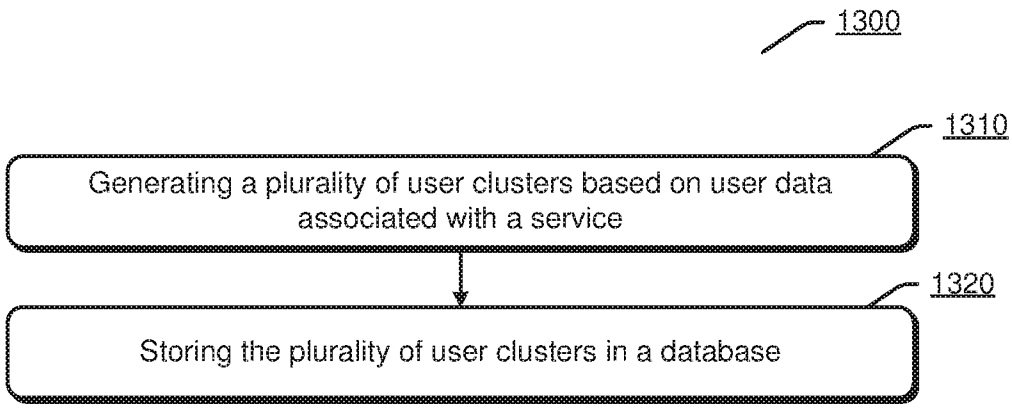
1300
1310
Generating a plurality of user clusters based on user data associated with a service
1320
Storing the plurality of user clusters in a database

FIG 13

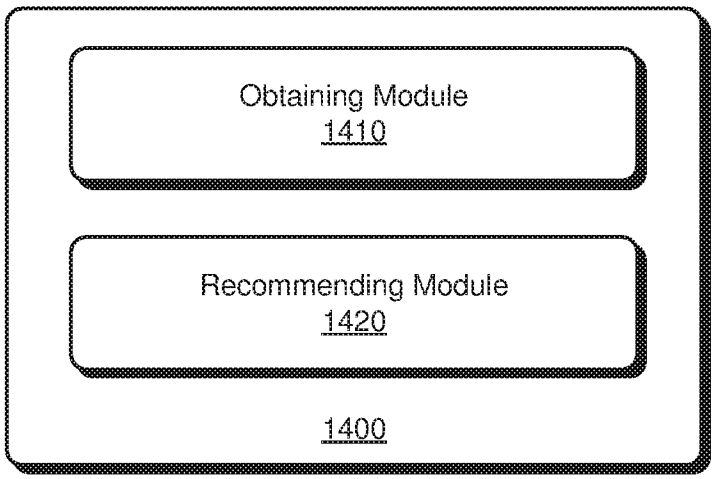

FIG 14

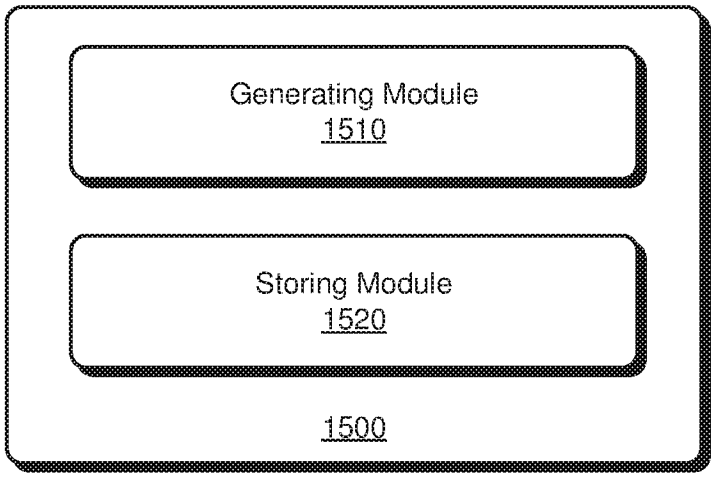

FIG 15

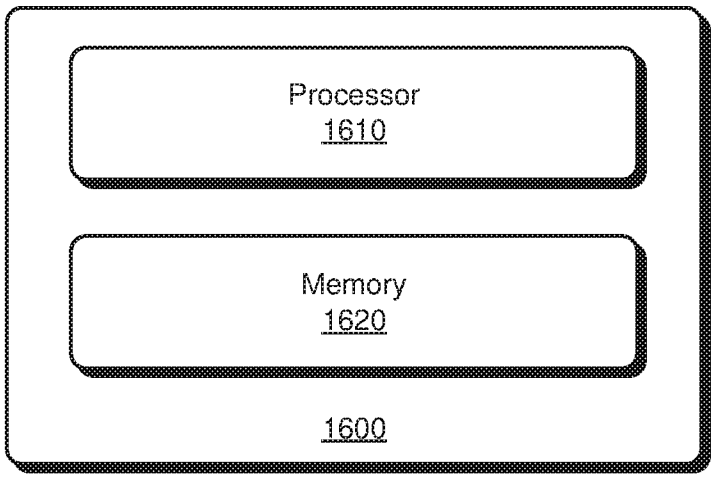

FIG 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2017/076133

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

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

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

EPODOC, WPI, CNPAT, CNKI, IEEE, GOOGLE: application, recommend, chat+, conversation, query, match, identify, match + rate?, activat+, switch+, response

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 105446994 A (HUAWEI TECHNOLOGIES CO., LTD.) 30 March 2016 (2016-03-30) Description, paragraphs [0051]-[0101]、[0109]-[0112], and figures 2-7	1-20
X	US 2014278343 A1 (TRAN BAO) 18 September 2014 (2014-09-18) Description, paragraphs [0030]-[0037]、[0076]-[0085], and figures 1、6	1, 12, 18
A	CN 104836720 A (BEIJING SAMSUNG COMMUNICATIONS TECHNOLOGY RESEARCH CO., LTD. ET AL.) 12 August 2015 (2015-08-12) the whole document	1-20
A	US 2015286709 A1 (SAMSUNG ELECTRONICS CO., LTD.) 08 October 2015 (2015-10-08) the whole document	1-20

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
05 December 2017	**22 December 2017**

Name and mailing address of the ISA/CN	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China**	**WU,Qing**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62413783**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2017/076133

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	105446994	A	30 March 2016	US	2016132605	A1	12 May 2016
				EP	3002693	A1	06 April 2016
				WO	2016004763	A1	14 January 2016
US	2014278343	A1	18 September 2014	US	2016034448	A1	04 February 2016
CN	104836720	A	12 August 2015		None		
US	2015286709	A1	08 October 2015	IN	201401782	I4	25 December 2015