May 31, 2019

On-device dialog management

Vinod D. Krishnan

Yanhong Chen

On-device dialog management

ABSTRACT

This disclosure describes techniques for on-device dialog management that enable a voice-based multi-turn dialog without the need for network connectivity. Although the techniques support a fully hands-free mode, parts of the dialog can also be entered by the user using a touch or other interface.

KEYWORDS

- Virtual assistant
- Interactive dialog
- Natural language understanding
- Speech recognition
- Hands-free
- Disambiguation

BACKGROUND

There are several situations where voice-based, hands-free interaction with a mobile device is advantageous to a user. Example applications of hands-free interaction include making phone calls; composing, sending or reading messages; accessing media; requesting navigation assistance; etc. while driving an automobile or using equipment. Effective hands-free interaction requires disambiguation in order to complete a task requested by the user, e.g., between multiple contacts who have the same first name, between multiple phone numbers for the same contact, etc. Robust hands-free interaction also needs to work with or without network connectivity and handle the situation where a user that starts a hands-free interaction could switch in mid-conversation to a non-hands-free, e.g., a touch interface, or vice-versa.

DESCRIPTION

The techniques of this disclosure enable a voice-based multi-turn dialog between a user and a virtual assistant to be serviced completely on-device. For example, the techniques enable a user to place a call using a conversation such as:

User: "Call John."

Virtual Assistant: "Sure. Which John?"

User: "John Baker."

VA: "Mobile or Work?"

User: "Mobile."

VA: "Thanks, now calling John Baker mobile."

Although the techniques support fully hands-free operation, user input provided in a non-hands-free manner, e.g., using a touch UI, can also be accepted.

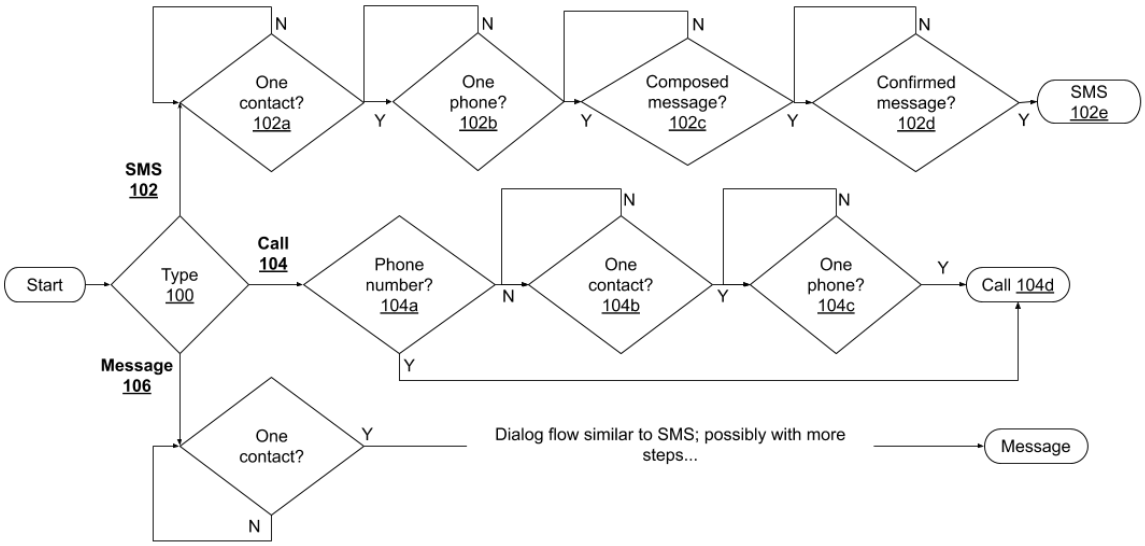

Fig. 1: Dialog management flow

Fig. 1 illustrates the flow of on-device dialog management, per techniques of this disclosure. The type of the user request is determined (100) and program flow proceeds along

one of a plurality of paths based on the type. For example, one possible path is towards composing an SMS (102). Another path is a path towards making a call (104). A third path is a path towards composing a message (106). Additional paths can be provided for additional functionality.

Within the SMS path, the name of the specific contact to send the SMS to is determined (102a) along with the specific phone (e.g., work, home, etc.) of the contact to send the SMS to (102b). The message is composed (102c), confirmed by the user as approved for sending (102d), and is sent (102e) thereafter. The message path (106) is similar to the SMS path, but can include additional steps, e.g., to select formatting options, message expiry options, etc. that are specific to the messaging modality.

Within the phone call path, the phone number to be called (104a) is determined and the call is placed(104d). In the absence of a phone number, the name of the specific contact to call (104b) is determined along with the specific phone to call (104c) to place the call.

Similarly, other paths can be defined for other functionality, e.g., for reading a message, which comprises a multi-step dialogue to pick out specific messages to be read and offers a reply option for each read message.

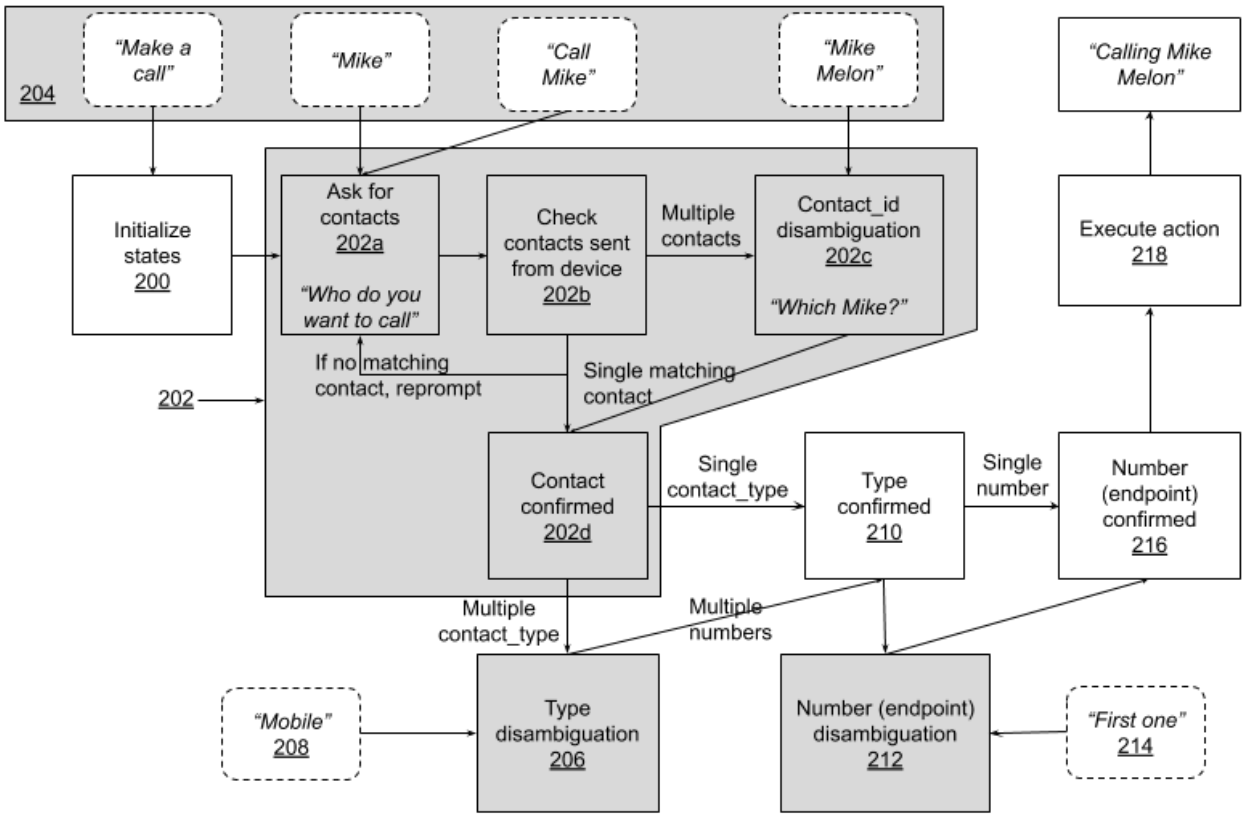

Fig. 2: Dialog flow and disambiguation to make a phone call

Fig. 2 illustrates the dialog flow and disambiguation to make a phone call. In Fig. 2, dotted rectangles with rounded corners represent speech input provided by the user. After initialization of states (200), dialog flow enters a contact-ID disambiguation phase (202), itself comprising loops wherein the user is requested for a contact to call (202a), contact-numbers determined from the device (202b), and contact-ID disambiguated (202c), leading to a confirmed contact-ID state (202d). User responses (204) guide the contact-ID disambiguation. In a similar manner, contact-type (e.g., mobile number, home number, etc.) is disambiguated (206) using user responses (208), and the phone number is disambiguated (212) using user responses (214). When dialog flow results in type-confirmed (210) and number-confirmed (216) states, the user-requested action, e.g., making a phone call, is executed (218).

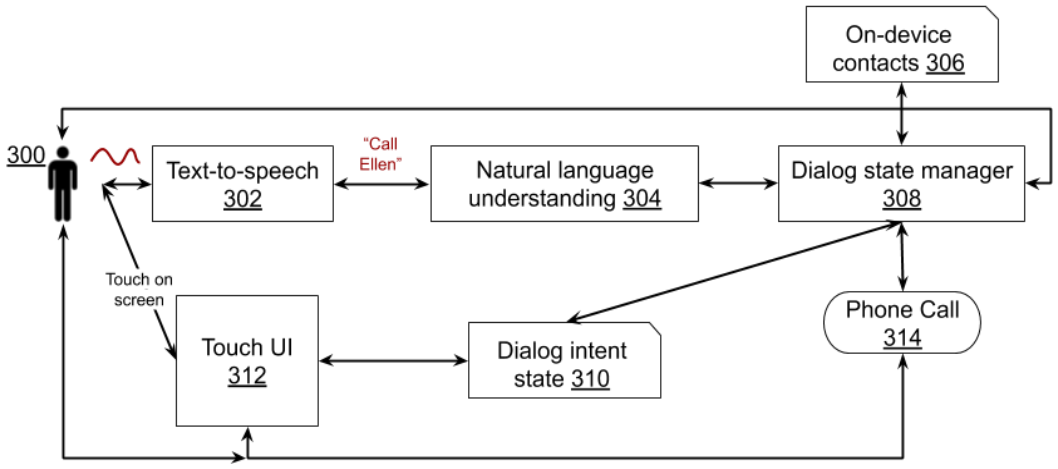

Fig. 3: Components of dialog manager

Fig. 3 illustrates the components of a dialog manager. With user permission, a text-to-speech component (302) transcribes the speech of the user (300). At any time during the conversation, the user can also interact with the dialog manager via a touch UI (312). A natural language understanding component (304) interprets the transcribed speech, e.g., as a request to make a phone call. The dialog state manager (308) maintains a form-like data structure that includes details of the action requested by the user. For example, if the user made a request to place a phone call, the dialog state manager maintains a data structure that includes name, phone number, type of phone, etc. The fields of the data structure are filled based on the conversation between the user and the dialog manager. The dialog intent state (310) includes fields that pertain to the user's immediately anticipated actions. Once the dialog state manager has gathered details pertaining to the user request, it performs the requested action, e.g., makes the phone call (314), drawing upon a database of user contacts (306) as necessary and permitted by the user.

Storing dialog state

A data structure is utilized to store information that pertains to the user request and forms the basis of disambiguation. This data structure is managed by the dialog state manager. Dialog

states are stored as fields within the data structure. During dialog, the states are filled in a manner similar to form-filling. The fields within the data structure comprises the states required for the fulfillment of a conversation, their values, their status, and the next field that the user is to be prompted for. The field status stores information relating to whether the user has been prompted for a field. The value status shows if a value for the field has been obtained from the user. If the user provides a partial value for a field, possible value candidates to the field are stored and the user requested to choose from the candidates. The dialog state can also be updated when the user interacts via non-hands-free, e.g., touch, interface.

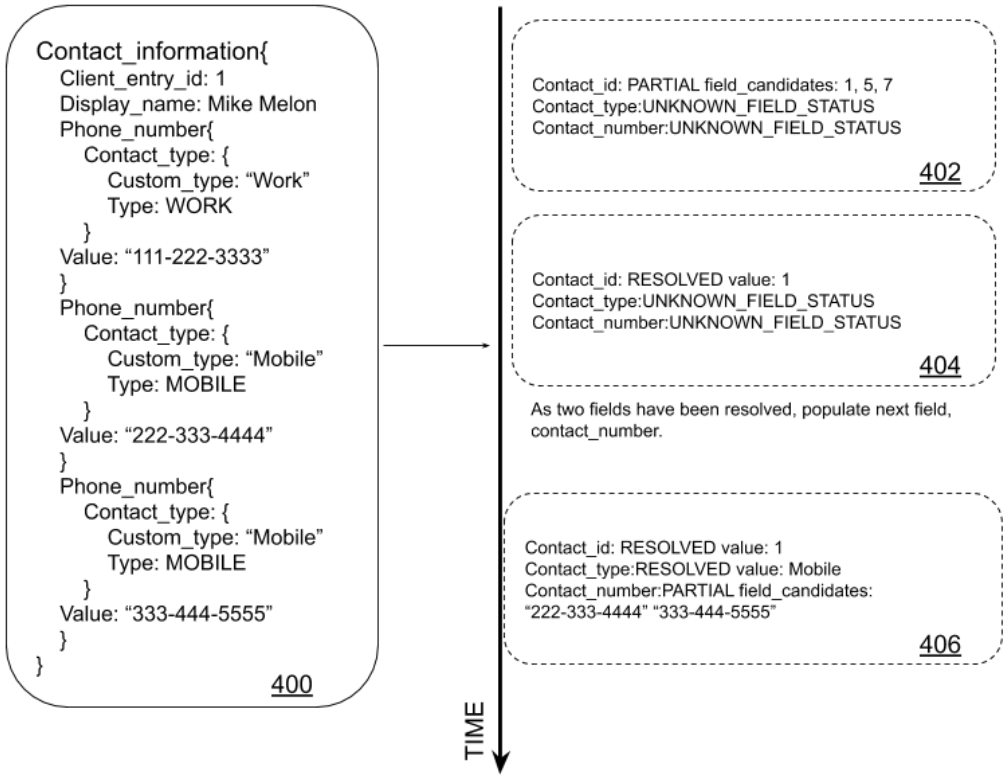

Fig. 4: Evolution of the dialog state with time

Fig. 4 illustrates the evolution of the dialog state as the conversation with the user proceeds. In this example, the user invokes a virtual assistant to place a call to the individual identified by the record 400. At the start of conversation, the dialog state (402) is filled with

relatively less information, e.g., mostly PARTIAL or UNKNOWN_FIELD_STATUS entries. With the passage of time and the continuing conversation with the user, more information is obtained, and the dialog state is filled with more information. For example, the dialog state at 404 comprises one RESOLVED field with known value, while the dialog state at 406 comprises multiple RESOLVED fields and no UNKNOWN fields.

Weighting spoken words in a user command

To correctly recognize a contact, interpretation of spoken words during the dialog can be performed with greater weight (bias) towards the sub-elements of the dialog. For example, if an initial section of the conversation indicated interest in a contact named John, and the user's contacts include "John Baker" and "John Taylor" in the contacts, interpretation of subsequent spoken queries can be performed such that words are more likely interpreted to be Baker or Taylor, rather than other similar sounding words, e.g., "bigger," "better," "Baylor," etc. The weighting can be based on context and dialog information.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of

a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes techniques for on-device dialog management that enable a voice-based multi-turn dialog without the need for network connectivity. Although the techniques support a fully hands-free mode, parts of the dialog can also be entered by the user using a touch or other interface.