November 2020

Two-stage Machine Learning Model for Local Processing of Frequent Queries

Vinod Das Krishnan

Two-stage Machine Learning Model for Local Processing of Frequent Queries

ABSTRACT

Virtual assistants implemented on a client device can recognize and fulfill spoken queries locally on device, e.g., using trained machine learning models. However, on-device query processing requires substantial processing power and memory. Heavy use of processor and/or memory resources can slow down the operation of the entire device, including queries for which processing is offloaded to an external server. This disclosure describes techniques to enhance the performance of local processing of voice based user queries via a two-stage serial pipeline of trained machine learning models. A first stage includes a limited-scope high efficiency model that can only recognize and fulfill the user's most frequent queries while a second stage includes the regular full-scope local query processing model that can recognize and fulfill a full range of queries.

KEYWORDS

- Virtual assistant
- Digital assistant
- Vehicle infotainment system
- Frequent queries
- Spoken command
- Voice command
- On-device query processing

BACKGROUND

When driving a vehicle, people often use a voice-based virtual assistant provided via a vehicle infotainment system, and/or via a mobile device such as a smartphone. Use of a voice-

based virtual assistant enables people to perform tasks, such as seeking navigation assistance, without having to take their hands off the steering wheel or needing to stop the vehicle in order to use the device to perform the task.

With user permission, voice-based virtual assistants typically utilize the device network connection to offload processing of the user's voice input to a remote server. The server performs query recognition and fulfillment. However, in situations such as when a user is driving (or is in a moving vehicle), a network connection of sufficiently high quality may not always be reliably available. To overcome the difficulties of unreliable network availability, virtual assistant applications can include the capability to operate by performing query understanding locally on the device and fulfilling the query, if feasible. Local query processing can match the quality and performance of remote query processing, especially for repeated queries which constitute the majority of user queries.

However, on-device query processing, e.g., that utilizes machine learning techniques, typically requires substantial processing power and memory, especially when the model is initialized. Such heavy use of processor and memory resources can slow down the operation of the device, including queries for which processing is offloaded to an external server. These issues are exacerbated when local query processing is used while using another resource heavy application, such as navigation applications that use digital maps.

DESCRIPTION

This disclosure describes techniques to enhance the performance of local processing of user queries issued to a voice-based assistant. The techniques involve processing the user's input via a two-stage serial pipeline of trained machine learning models. Specifically, the first stage includes a smaller, efficient model that is trained to only recognize the user's most frequent

queries while the second stage includes a regular, larger model used that can perform local query processing for the full range of user queries.

Because of the limited scope (recognizing frequent queries), the first-stage model is smaller and simpler, and requires substantially lower memory and processing resources to fulfill the user queries it is trained to support. As a result, the first-stage model can be kept loaded in memory (e.g., similar to a hotword detection model) and can provide performance enhancement for local processing of the user's frequent queries.

If the user's query is not recognized by the first-stage model (it is different from frequent requests), it is passed along to the second-stage model for processing as normal. If processing the query locally via the second-stage is prohibitively resource-heavy or infeasible (e.g., the second model fails to recognize or fulfill the query) then query processing can be offloaded to a remote server as normal, if the user permits server-side query processing.

Appropriate feedback mechanisms are implemented to pass relevant information from the second-stage model to the first-stage model in order to improve the recognition performance of the first-stage model. In order to maintain the limited scope and small memory and processing footprint of the first-stage model, with user permission, the set of frequent user queries is periodically refreshed (e.g., using the feedback from the first-stage model) to remove the support for queries that are used less frequently over time and/or add support for newer frequent queries.

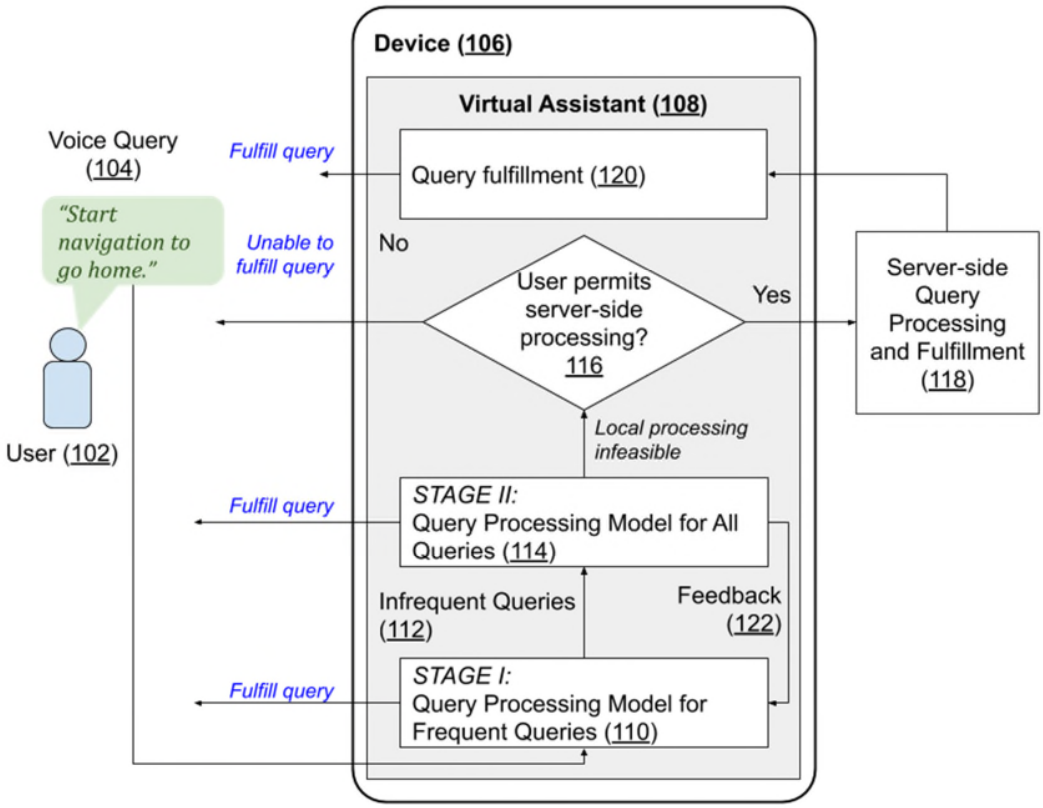

Fig. 1: Two-stage serial pipeline of machine learning models for local query processing

Fig. 1 shows an example operational implementation of the use of two different models for local query processing. A user (102) issues a voice query (104) to a virtual assistant (108) provided via a device (106). With user permission, the first-stage query processing model trained to recognize and process the user's frequent queries (110) determines whether the current query (in this case, starting the navigation to go home from the current location) is one of the supported frequent queries. If the query is supported, it is fulfilled via local processing using the first-stage model.

Queries that are not supported or understood by the first-stage model, e.g., infrequent queries (112) are passed to the second stage model (114). If local processing is appropriate and feasible, the second-stage model performs local processing to fulfill the queries that were not

served by the first-stage model. If local processing is not appropriate or feasible, query processing and fulfillment is performed using standard server-side mechanisms (118) if the user permits such processing (116) and the results are returned to the user via a query fulfillment module (120). If the user permits, feedback from the second-stage model (122) is used to improve the performance of the first-stage model and update the set of frequent queries it supports.

With user permission, the first-stage model can be seeded to handle the user's top N frequent queries, where N is a parameter that can be set by the developers and/or specified by the user and/or determined dynamically at runtime. The set of top queries can be limited to be within a given usage scenario, such as driving.

The described techniques can be implemented on any device that provides voice-based virtual assistant functionality with local query processing capabilities. Implementation of the techniques can improve the performance of on-device query processing, thus enhancing the user experience (UX) of using voice-based virtual assistants. Such enhancements can be particularly useful in resource-constrained usage scenarios, such as driving.

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 frequent queries, a user's context, 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 to enhance the performance of local processing of voice based user queries via a two-stage serial pipeline of trained machine learning models. A first stage includes a limited-scope high efficiency model that can only recognize and fulfill the user's most frequent queries while a second stage includes the regular full-scope local query processing model that can recognize and fulfill a full range of queries. With user permission, the first stage model can be cached in memory and can provide significant performance improvements for frequent queries. The described techniques can be incorporated within any device that provides voice based virtual assistant functionality with local processing capabilities.