October 2020

Early Abandonment of On-device Query Processing When Fulfillment is Deemed Infeasible

Matthew Sharifi

Victor Carbune

Early Abandonment of On-device Query Processing When Fulfillment is Deemed Infeasible

ABSTRACT

Devices that provide voice assistant features can handle voice commands or spoken queries locally on device or by offloading the processing to an external server. Local query processing can eliminate the latency and potential unreliability of server-side processing. When local processing and fulfillment is infeasible or unlikely, query fulfillment is performed on a server, if permitted by the user. In such a case, the computation involved in the unsuccessful attempt at local processing consumes local resources without a usable result. This disclosure describes techniques to determine locally if a spoken query cannot be served by the device and abandoning local processing upon such a determination. The user is informed of the query abandonment, and if the user permits, server-side query processing is carried out to provide a response to the query.

KEYWORDS

- Spoken query
- Voice command
- Query fulfillment
- Query abandonment
- Query prediction
- On-device query processing
- Server-side query processing
- Virtual assistant
- Voice assistant

BACKGROUND

Voice-based virtual assistants enable people to engage in natural conversational interaction to obtain information and/or perform actions. Users interact with virtual assistants via various devices, such as smartphones, smart speakers, smartwatches, etc. Many such devices handle voice commands by offloading the processing to a server external to the device. However, such server-side processing can result in increased latency and decreased reliability along with bandwidth costs,, thus having a negative effect on user experience (UX). Moreover, some users may prefer settings that require processing of the query on the local device and prevent transmission of the query to a server.

To overcome the shortcomings of server-side processing of queries issued to a virtual assistant, it is possible to process queries locally on the device itself. However, many queries cannot be handled locally on the device, e.g., owing to device limitations and/or the type or content of the query itself. In such cases, it becomes necessary (if permitted by the user) for the query processing to fall back on server-side processing for interpretation of the query and/or generation of a response. Needing to use server-side processing as fallback in such a situation can introduce additional latency because of the time required for the initial unsuccessful attempt at local device processing. In addition, the unsuccessful attempt at local processing of the entire user query imposes high computational costs involved in automated speech recognition (ASR) and natural language understanding (NLU). These wasted computational cycles can consume power and drain the device battery (when the device that receives the query is a battery-powered device such as a smartphone or smartwatch), thus degrading the overall user experience, especially on low-power devices such as wearables.

DESCRIPTION

This disclosure describes techniques for early stopping of local on-device processing of spoken queries issued to a virtual assistant on the device. Users can issue a spoken query to the virtual assistant as they normally do, with the virtual assistant being invoked using interaction techniques such as an activation hotword, gesture, button press, etc. Query processing is attempted locally on the device by beginning to perform the needed computation, such as ASR, NLU, etc. However, the attempt to process the user's voice query on the device is abandoned without waiting for the entire query to finish processing as soon as it is determined that the query cannot (or likely cannot) be handled with local processing.

As the local query processing on the device proceeds, a set of partial query hypotheses is generated at suitable intervals, such as after processing each word spoken by the user. The hypotheses represent predicted possible queries based on the query input processed until that point. Each hypothesis is associated with a corresponding confidence score that indicates the likelihood of the hypothesis predicting the correct query. Each time a new set of hypotheses and corresponding confidence scores is generated, the top hypotheses within the set are parsed to determine the likelihood that each hypothetical query can be processed and fulfilled locally on the device. If no hypothesis is associated with a sufficiently high likelihood that it can be fulfilled locally, all on-device query processing operations are stopped immediately and optionally, the user is informed that the query cannot be handled locally on the device. If the user settings are such that server-side query processing is restricted, the operation terminates at this point. Alternatively, the user can be offered the option to process the query server-side along with an indication that handling the query via on-device local processing is infeasible.

If the user settings permit server-side query processing, the spoken query can be additionally processed on the server side in parallel with the on-device local processing, or upon user selection of the option for server-side query processing. In case on-device processing is determined to be infeasible, the query processing can switch automatically to server-side operation.

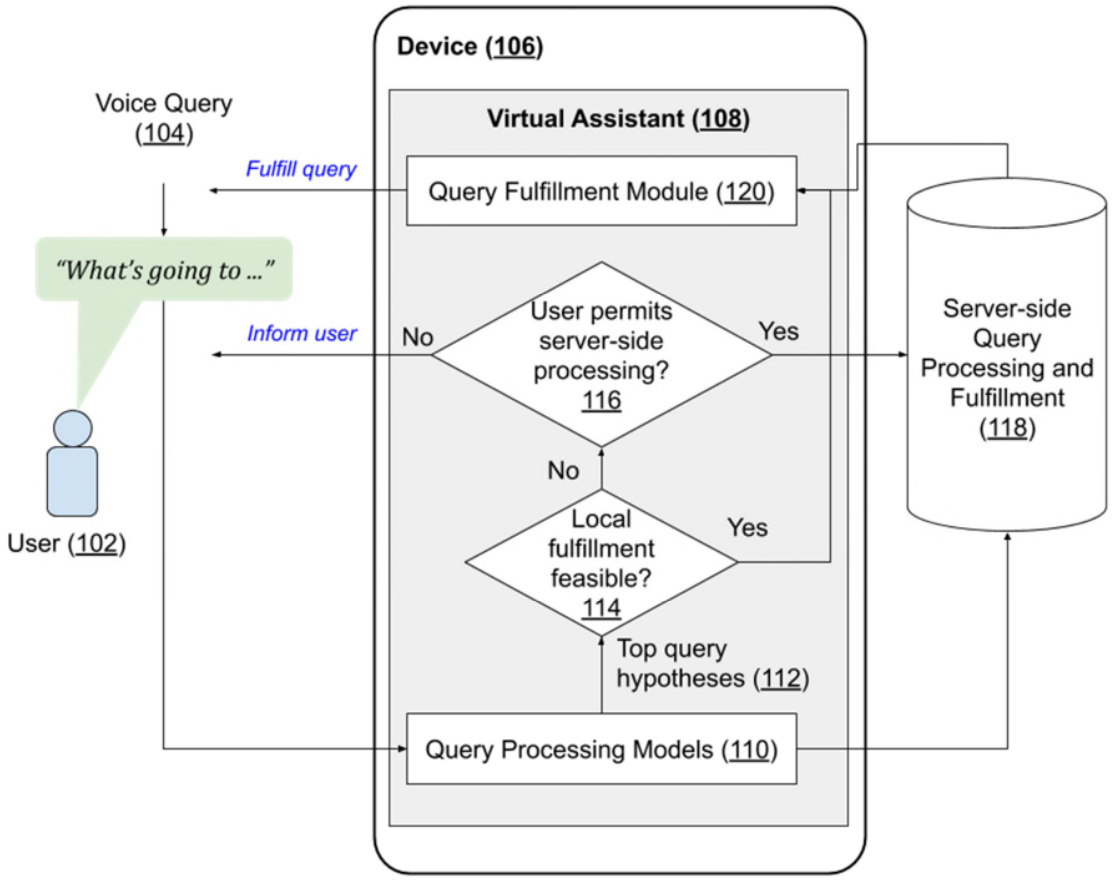

Fig. 1: Abandoning on-device query processing when local processing is deemed infeasible

Fig. 1 shows an operational implementation of the techniques described in this disclosure. A user (102) starts to issue a voice query (104) to a virtual assistant (108) provided via a user device (106). On-device query processing models (119) are used to generate a set of hypotheses regarding the likely query as the query is being received. The top query hypotheses (112) are

analyzed to determine if they can be fulfilled locally on the device. If none of the hypotheses are deemed to be suitable for fulfillment (114) via local processing on the device, on-device processing is stopped immediately even if the query is not yet completely issued and/or processed, and the user is informed accordingly. Otherwise, the query is fulfilled via local processing by a query fulfillment module (120) on the device after the entire query is received. If the user permits (116), the query can be relayed for server-side processing (118) external to the device. With user permission, the query can be fulfilled by server-side processing, which can serve the answer to the user via the query fulfillment module, whenever fulfillment via local on-device processing is deemed infeasible.

The generation of the set of partial query hypotheses can be performed via standard query parsing techniques, with models capable of operating on partial or incomplete queries using one or more suitable techniques, such as grammars, machine learning models, prefix-based matching, etc. The top hypotheses from the generated set can be selected based on an appropriate criterion, such as a specific number, a confidence score threshold, etc. The likelihood that a hypothesis can be processed locally on the device can be derived by combining the confidence score associated with local processing feasibility combined with the confidence score indicating the likelihood that the hypothesis represents the correct user query.

The various threshold values used in the above operation can be set by the developers and/or specified by the user and/or determined dynamically at runtime. Moreover, the threshold values can differ based on whether fallback server-side processing is feasible or in-progress in parallel. If fallback server-side processing is feasible or already in-progress, the threshold values can be higher because the availability of the server-side option avoids the user disruption caused by early stopping of the attempted local on-device processing. In contrast, if server-side

processing is infeasible or not permitted by the user, lower threshold values can be chosen to ensure that the attempt to handle the query on the device is abandoned only when there is high certainty that local processing is infeasible.

By stopping on-device query processing attempts at the earliest possible stage, the described techniques can save processing resources, such as processor time, memory, power, etc. The resource savings avoid unnecessary drain on the device battery, thus allowing the device to be usable longer without needing to be recharged. Further, informing the user as soon as it is determined that local query processing is infeasible saves time in proceeding with a longer query that is ultimately unsuccessful. These benefits serve to enhance the user experience of voice-based virtual assistants that support local on-device query processing and fulfillment.

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 spoken queries, query history, 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

Devices that provide voice assistant features can handle voice commands or spoken queries locally on device or by offloading the processing to an external server. Local query processing can eliminate the latency and potential unreliability of server-side processing. When local processing and fulfillment is infeasible or unlikely, query fulfillment is performed on a server, if permitted by the user. In such a case, the computation involved in the unsuccessful attempt at local processing consumes local resources without a usable result. This disclosure describes techniques to determine locally if a spoken query cannot be served by the device and abandoning local processing upon such a determination. The user is informed of the query abandonment, and if the user permits, server-side query processing is carried out to provide a response to the query.