November 2020

Contextually Performing Query Processing On-device or On a Remote Server

Vinod Das Krishnan

Vikram Aggarwal

Contextually Performing Query Processing On-device or On a Remote Server

ABSTRACT

When a user issues a query, e.g., a spoken query to a user device such as a smartphone, smart speaker, in-car device, etc., the query may be processed locally on-device and additionally, remotely on a server (if permitted by the user). The determination of whether a query is processed locally or on a remote device is typically based on whether the device has a network connection. Local processing of queries can consume device resources. When the device is simultaneously in use for other critical tasks, such resource demand can have a negative impact on such tasks. This disclosure describes the use of a trained machine learning model that takes into account user-permitted contextual factors to determine whether query processing is to be performed on-device or on a remote server.

KEYWORDS

- Virtual Assistant
- On-device query processing
- On-device operation
- User context
- Device resources
- Network latency

BACKGROUND

When a user issues a query, e.g., a spoken query to a user device such as a smartphone, smart speaker, in-car device, etc., the query may be processed locally on-device and additionally, remotely on a server (if permitted by the user). Some queries may be of a nature that can be fully serviced on-device. The determination of whether a query is processed locally or on a remote device is typically based on whether the device has a network connection. For example, if the device operating system reports that no network connection is available, query processing is attempted locally. When a network connection is available, query processing can be performed in parallel - (a) locally on-device; and (b) on a remote computer. The earliest response available is then served to the user.

Local processing of queries can consume substantial amounts of device resources, such as processor time, memory, battery, etc. When the device is simultaneously in use for other critical tasks, such resource demand can impact such tasks. For example, when a smartphone is paired to an in-car display and provides navigation assistance, on-device query processing can have a negative impact on the navigation assistance task. To avoid the negative impact on such other tasks, the on-device query processing stack (which may be processor and/or memory intensive may be implemented with a delay.

A binary decision based on whether a network connection is available is insufficient for optimal query processing. A static algorithm that makes such a determination can fail due to constantly changing parameters such as device hardware capability, current usage, battery level, etc. Such an algorithm is also expensive to build, since it requires implementation of custom logic per configuration to ensure that it supports different current and future hardware configurations.

DESCRIPTION

This disclosure describes the use of a trained machine learning model to determine whether query processing is to be performed on-device or remotely (if permitted by the user). The machine learning model takes as input various user-permitted contextual factors, as detailed below.

1. **Device-specific information:** device region, Mobile Country Code (MCC), Mobile Network Code (MNC), Region, etc.
2. **Contextual information:** is the device stationary or moving (e.g., in a car) based on accelerometer or other sensors, device location (e.g., based on a location sensor)
3. **Network information:** latency (e.g., time it takes to establish a connection to a remote query-processing server), data throughput (e.g., as measured by the server that receives the query), time to obtain results from the server, etc.
4. **Information specific to virtual assistant:** the sign-in status of the device (signed-in devices may use a different server code path than devices that are signed-out), whether user data is used to service the query (e.g., the user's contacts), data sent to the server during the query (e.g., information regarding the capabilities of the user's vehicle), version information of the virtual assistant and/or on-device machine learning model(s) used for query processing, grammar capabilities for on-device processing, etc.
5. **Hardware information:** whether the device includes a graphical processing unit (GPU) or a processor optimized for speech recognition and/or machine learning; availability status of the device processor and/or memory at the time of query, available disk space on the device (e.g., necessary to store language packs which may be used for on-device query processing).

The contextual factors as described above are accessed and provided to the trained machine learning model with specific user permission. Users are provided with information indicating that the model may use one or more such factors to enhance query processing decisions, and are provided with options to disable access to one or more of the contextual factors. If the user denies permission for use of contextual factors, query processing is performed in a conventional manner, e.g., based on network availability status, and in parallel on the server and on-device.

If the user permits access to contextual factors, the trained machine learning model makes a determination regarding whether to process the query on-device or to have a remote server process the query. For example, if the input factors indicate that the device processor is busy (e.g., serving a critical task such as providing navigation assistance, or performing other processor-intensive activities such as media playback), the model may output an indication that the query is to be processed on the remote server. The use of a ML model ensures that the varying input factors (which may differ for different devices) and their values (which may vary widely based on context) are taken into account without writing custom logic. Further, if the user permits, the model can learn from prior query processing decisions and improve over time.

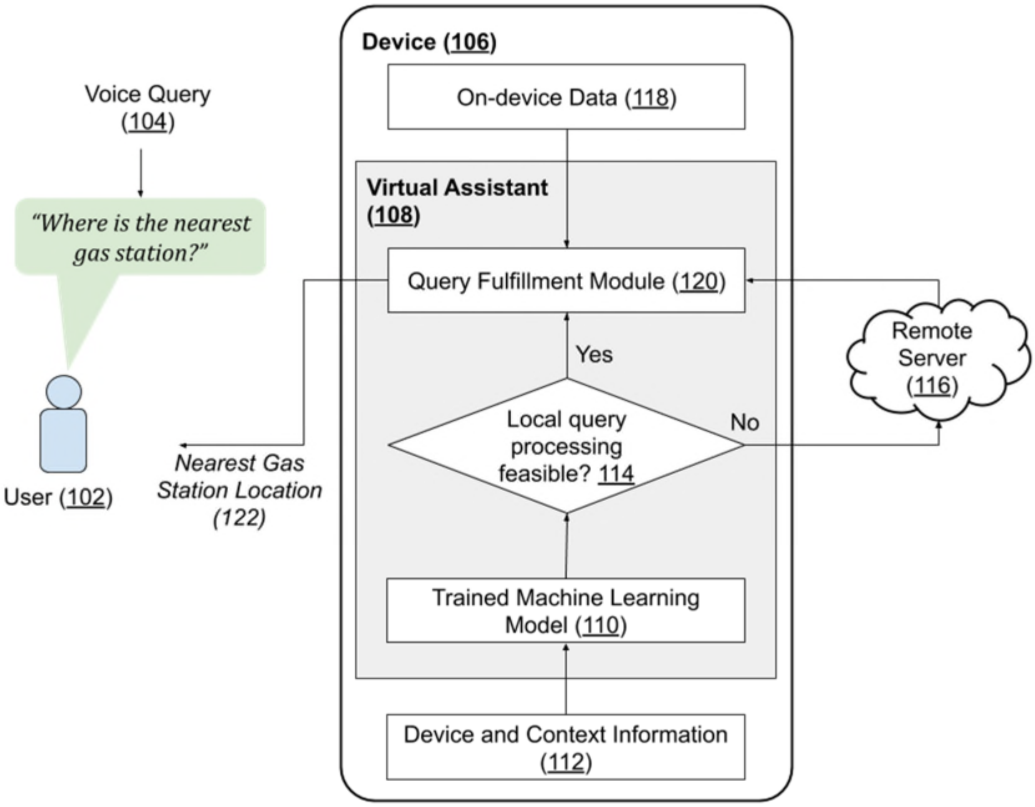
Voice Query (104)
"Where is the nearest gas station?"
User (102)
Nearest Gas Station Location (122)
Device (106)
On-device Data (118)
Virtual Assistant (108)
Query Fulfillment Module (120)
Yes
Local query processing feasible? 114
No
Remote Server (116)
Trained Machine Learning Model (110)
Device and Context Information (112)

Fig. 1: Contextually Performing Query Processing On-device or On a Remote Server

Fig. 1 shows an example of operational implementation of the techniques described in this disclosure. A user (102) issues a voice query (104) to a virtual assistant (108) on a user device (106). In this example, the device is a smartphone that is connected to an in-car display and is providing navigation assistance. Based on user-permitted contextual information (112), a trained machine learning model (110) is used to determine whether the query is to be processed on-device or on a remote server.

If on-device processing is feasible (114), the query fulfillment module (120) serves the query via on-device processing utilizing data available on the device (118). On the other hand, if the on-device mode is determined to be infeasible and remote query processing is feasible (a sufficiently high quality network connection to a remote server is available), the query fulfillment module obtains the pertinent information from a remote server (116). The query response is provided to the user.

For example, if the device has low available computational capacity (e.g., due to the navigation task taking up processor and memory), the ML model (110) may make a determination to offload query processing to the remote server, while if the device has significant local computational capacity (e.g., has a dedicated and available GPU for query processing), the ML model may make a determination to perform query processing on-device.

The techniques can be particularly useful in situations with frequently changing external conditions, such as when the user is driving (the device location is changing frequently and the device is being used for navigation) and interacting with a virtual assistant via voice. The techniques described herein enable nuanced decision making regarding query servicing, thus facilitating a smooth user experience while simultaneously optimizing device resource utilization.

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 context such as device resource availability, 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 the use of a trained machine learning model that takes into account user-permitted contextual factors to determine whether query processing is to be performed on-device or on a remote server. The use of a ML model eliminates the need for custom logic for different device configurations and can optimally account for a variety of contextual factors that can affect query processing. Also, the ML model can improve over time based on observed query performance, if permitted by the user. The techniques described herein enable nuanced decision making regarding query servicing, thus facilitating a smooth user experience while simultaneously optimizing device resource utilization.

REFERENCES

1. Bringert, Bjorn Erik, Johan Schalkwyk, Michael J. Lebeau, Richard Zarek Cohen, Luca Zanolin, and Simon Tickner. "Hybrid Client/Server Speech Recognition In A Mobile Device." U.S. Patent Application 13/586,696, filed August 15, 2012
2. Aggarwal, Vikram, and Moises Morgenstern Gali. "Learning offline voice commands based on usage of online voice commands." U.S. Patent Application 15/862,615, filed January 4, 2019.