October 2019

Providing Real-Time Captured Information Input to a Machine-Learned Model to Improve Query Services

Perumaal Shanmugam

Felipe Leme

Julian Odell

Providing Real-Time Captured Information Input to a Machine-Learned Model to Improve Query Services

Abstract:

This publication describes techniques and methods that a computing device uses to provide improved query services (*e.g.*, autofill suggestions, speech biasing for automatic speech recognition) to applications on the computing device. To this end, an information collector on the computing device collects application activity information, information displayed on a display, and event information. This collected information can be provided as input to a machine-learned model implemented on the computing device. Responsive to the input received, the machine-learned model can classify the collected information to determine relevant attributes (*e.g.*, keywords, searched locations, names) and make suggestions for utilization by query services provided by the computing device. Through these techniques and methods, user privacy is maintained, less power is consumed by the computing device, and the resources of the computing device (*e.g.*, memory) are conserved.

Keywords:

Computing device, smartphone, machine-learning, query services, autofill, autocomplete, autosuggest, content capture, entity, context, auto-speech recognition (ASR), query completion, query suggestion, voice command services, suggestion

Background:

Widely used computing devices (*e.g.*, smartphones, tablets, laptops) enable users to perform many actions, such as texting, emailing, searching for information on the web, and searching for a location on a digital map. Frequently, to perform these actions, users need to provide input (*e.g.*, type words on the keyboard, utilize a user interface) to the computing device. For user convenience, the computing device provides query services (*e.g.*, autofill suggestions, speech-to-text recognition (speech biasing) utilized by automatic speech recognition services) to applications on the computing device. The term “autofill,” refers to a feature in which the computing device (*e.g.*, the operating system (OS), an application) makes suggestions for filling in data input fields based on objects stored in a “dictionary.”

To provide query services, the computing device may log commonly used words in a dictionary and suggest these words based on the likelihood of reusage. For example, a user (John) searches a mapping service on his smartphone for a certain restaurant. Additionally, John checks his calendar application to verify he has no appointments during his lunchtime. Upon locating a desired restaurant and checking his calendar, John texts a friend (Jacob) to see if he would like to go out to eat. While typing the text message (*e.g.*, “Do you want to eat at”) to Jacob, John’s smartphone may present John with many word choices, such as “What,” “To,” or “I,” providing a number of regularly used words as suggestions. What John may desire to see as a suggestion is the name of the recently searched restaurant or the available time slot in his calendar. However, John may not desire to have the mapping service have access to his calendar data for privacy reasons, nor may he desire the battery power and resources (*e.g.*, random access memory) drain that would result from having the mapping service access his calendar data.

Description:

This publication describes techniques and methods that a computing device can perform to provide improved query services (*e.g.*, autofill suggestions, speech biasing for automatic speech recognition) to applications on the computing device.

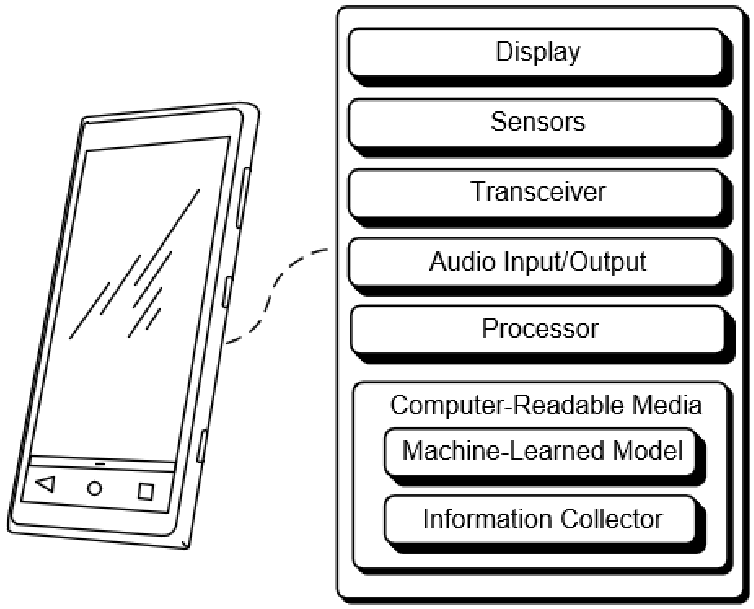

Figure 1

Figure 1, above, illustrates an example computing device and elements of the computing device that support the techniques and methods described in this publication. As illustrated in Figure 1, the computing device is a smartphone. However, other computing devices (*e.g.*, tablets, laptops) can also support the techniques and methods described in this publication. The computing device includes a display, a transceiver for transmitting data to and receiving data from the access point of the wireless network), a processor, and an audio input/output device (*e.g.*, a microphone).

In addition, the computing device includes a computer-readable medium (CRM) storing executable instructions. The CRM includes a machine-learned model (ML model). The ML model may be a standard neural-network-based model with corresponding layers required for processing input features like fixed-side vectors, text embeddings, or variable length sequences. The ML model is trained to classify collected information to determine relevant attributes and make suggestions for query services on the computing device.

The CRM further stores executable instructions of an information collector. The information collector continuously generates real-time information relating to application activity, information displayed on a display (display information), and information relating to events (collectively "collected information"). Collected information could include information within on-screen entities, application context information, and information displayed on the display of the computing device. The information collector, when executed by the processor of the computing device, causes the computing device to perform operations described in this document.

The collected information is provided to the ML model as input. The ML model classifies the collected information to determine relevant attributes (*e.g.*, keywords, searched locations, names) and makes suggestions for utilization by query services provided by the computing device. By only providing relevant suggestions (instead of the collected information in general), the computing device controls access to the collected information, thereby preserving user privacy. Further, by providing controlled access to relevant collected information, applications on the device do not need to separately collect, store, and process such information on an application-by-application basis, thereby preserving the resources of the computing device and conserving battery power.

For example, a user (John) receives an email indicating a required conference on the 20th day of the next month at 2:00 p.m. John opens the email, reads it over, and desires to schedule the event in a separate calendar application. So, he opens the calendar application, creates a new event, and begins typing details from the email, such as the name of the conference, the time, and the location. While John was performing all these actions, the information collector was operating in the background of the computing device and captured information that was displayed in the email application. The collected information was then inputted into the ML model, which identified relevant attributes and provided suggestions to the computing device for use with query services. The query services were then updated with the suggestions made by the ML model to include keywords based on the collected information. Thus, while John is typing details of the conference into his calendar application, the computing device makes autofill suggestions that include words or sentences related to the conference.

Query services can also include speech-to-text recognition (speech biasing) utilized by automatic speech recognition (ASR) services. Some users have accents, while other users use different terminology when speaking. As a result, automatic speech recognition (ASR) services may sometimes have difficulty interpreting user voice commands. The techniques and methods described in this publication can be utilized to improve speech-to-text recognition (speech biasing). For example, collected information can be inputted to the ML model, the ML model can determine relevant attributes and make suggestions for speech-to-text recognition to facilitate the interpretation of user voice commands. For example, John is looking at an address displayed on the display of the computing device. John performs a voice command stating, "Navigate me there," however the ASR services is unable to understand the word "there" in John's voice command. Collected information related to information displayed on the display can be inputted into the ML model, the ML model can classify the collected information to determine relevant attributes (*e.g.*, identify an address), and the ML model can make suggestions for the ASR services enabling the interpretation of John's voice command.

Furthermore, the techniques and methods described in this publication can be utilized to improve referential query completion. Referring to the prior example, John is looking at an address displayed on the display of the computing device. John performs a voice command stating, "Navigate me there." Collected information related to information displayed on the display can be inputted into the ML model, the ML model can classify the collected information to determine relevant attributes (*e.g.*, identify the address) and makes suggestions for query services on the computing device (*e.g.*, the identified address is the desired location).

Furthermore, the techniques and methods described in this publication can be utilized to improve contact query completion. As another example, if John speaks a voice command to a virtual personal assistant instructing the computing device to "Call Tom," collected information (*e.g.*, information related to communications by John with his contacts) can be inputted into the ML model, the ML model can classify the collected information to determine relevant attributes (*e.g.*, contacts named "Tom") and make suggestions for utilization with autofill suggestions on the computing device (*e.g.*, which Tom among the many Toms in John's contact list John intends to call). An illustration of the computing device determining the correct contact can be seen in Figure 2.

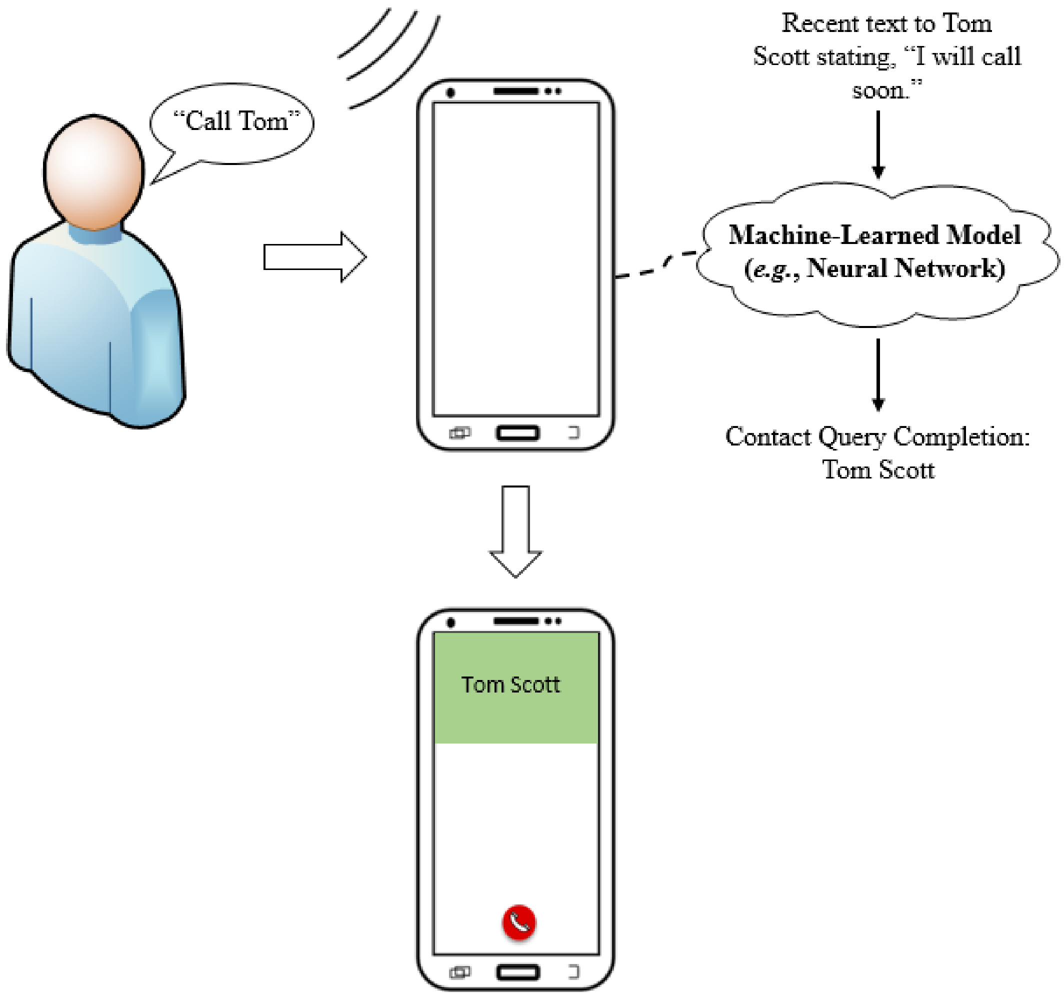

Figure 2

As illustrated in Figure 2, the computing device is a smartphone. John had recently texted his friend Tom Scott stating that he would call soon. Information related to the content of the text message was collected by the information collector, relevant attributes were determined by the ML model, and the ML model made a suggestion for query services on the computing device, enabling a virtual personal assistant to initiate a phone call with the appropriate Tom in John's contact list.

By only providing relevant suggestions (instead of the collected information in general) to other applications via query services, the computing device controls access to the collected information, thereby preserving user privacy. Further, by providing controlled access to relevant collected information via query services, applications on the device do not need to separately collect, store, and process such information on an application-by-application basis, thereby preserving the resources of the computing device and preserving battery power.

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), 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.

In conclusion, the techniques and methods described herein, where an information collector continuously generates real-time captured information, a ML model receives this collected information and determines relevant attributes, and the ML model makes suggestions for utilization by query services, maintains user privacy, consumes less power on the computing device, and conserves computing device resources.

References:

[1] Adam Rodriquez, Rendong Chen, Thomas Horn, Florbela Lei, Yohan Launay, Chen-Ting Chuang, and Bin Zhu. Suggested Items for Use with Embedded Applications in Chat Conversations. US Pub. 2018/0367484, filed June 15, 2017, and published December 20, 2018.

[2] Matthew Sharifi and David Petrou. Detection and Ranking of Entities from Mobile Onscreen Content. US Patent 9788179, filed May 14, 2015, and issued October 10, 2017.