February 2021

On-Device Query and Metadata Caching for Expedited Inference and Rendering of Answer Cards

Keun Soo Yim

On-Device Query and Metadata Caching for Expedited Inference and Rendering of Answer Cards

Abstract:

This publication describes methods and techniques for mobile devices to enhance the user experience associated with search applications. In aspects, an Intent Manager associates a query suggestion to a corresponding answer card (e.g., search result) for a given query text. The association may involve the Intent Manager identifying that the submission of a query suggestion results in a search application presenting one or more corresponding answer cards. Further, the association may include the production of metadata relating the query text to the submitted query suggestion. The Intent Manager can then cache the metadata on the device. In doing so, when a user performs an identical search at a later time, the Intent Manager can search through the cached metadata for a query text, find the query text, and extract the related query suggestion from the metadata. Using the related query suggestion as the search criteria, the search application can simultaneously present a query suggestion and its corresponding answer card(s) to a user during a search.

Keywords:

Caching, querying, metadata, computing device, prefix, predictive text, autofill, query suggestion, remote procedure call (RPC), rich user interface (UI), personal query, query text, suggestion card, answer card, personal dictionary, personal corpus, public corpus

Background:

Users frequently search for and discover content using search applications on their mobile devices (e.g., smartphones, laptops). For example, a user may use a search application on his mobile device to determine the weather in San Jose, California. To enhance the search experience, the mobile device may store textual searches ("personal queries") in a personal query set (e.g., personal dictionary) and later provide these personal queries as query suggestions (e.g., predictive text, auto-suggest) the next time the user begins a search. Continuing the above example, the mobile device may store the personal query ("Weather San Jose") so the search application can present the personal query as a query suggestion at another time. As a result, if the user later begins typing "Wea" in the search application, then the mobile device can provide "Weather San Jose" as a query suggestion. The user may then select the query suggestion to view the answer (e.g., search result) to the search.

Search applications may present an answer to a search on a user interface (UI) of the device. For example, some search applications present an answer using a card-based user interface (answer card UI). Figure 1, below, illustrates a search application utilizing an answer card UI to answer a user query with weather information for San Jose, California.

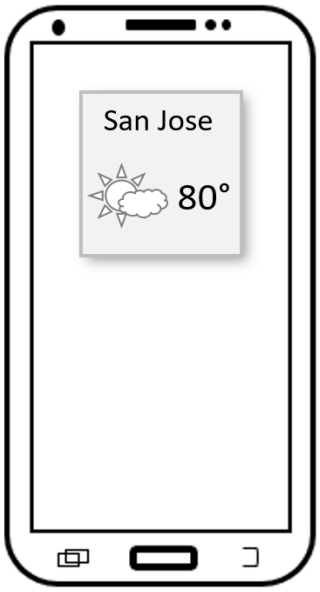

Figure 1

As illustrated, an answer card UI presents the answer to a search in an answer card (e.g., a rectangular graphical element containing a search result to a query). Utilizing an answer card UI enables a user to quickly identify an answer to their query in an aesthetic layout. To further enhance user experience, it is desired to simultaneously present query suggestions and their corresponding answer cards. In this way, users can view answer cards without having to select query suggestions and, therefore, discover content more quickly.

Description:

This publication describes methods and techniques for mobile devices to enhance the user experience associated with search applications. In aspects, an Intent Manager associates a query suggestion to a corresponding answer card for a given query text. The association may involve the Intent Manager utilizing an algorithmic (e.g., machine-learned) technique configured to identify that the submission of a query suggestion results in a search application presenting one or more corresponding answer cards. Further, the association may include the production of metadata

relating the query text to the submitted query suggestion. The Intent Manager can then cache the metadata on the device. In doing so, when a user performs an identical search at a later time, the Intent Manager can search through the cached metadata for a query text, find the query text, and extract the related query suggestion from the metadata. Using the related query suggestion as the search criteria, the search application can simultaneously present a query suggestion and its corresponding answer card(s) to a user during a search.

While the example mobile device described in this publication is a smartphone, other types of mobile devices may also support the methods and techniques described herein. A mobile device may include one or more processors, transceivers, an input/output device (e.g., a display panel), and a computer-readable medium (CRM). The CRM may include any suitable memory or storage device like random-access memory (RAM), static RAM (SRAM), dynamic RAM (DRAM), non-volatile RAM (NVRAM), read-only memory (ROM), or flash memory. The CRM includes device data (e.g., user data, multimedia data, applications, and/or an operating system of the device), which are executable by the processor(s) to enable the techniques and methods described herein. The device data may include an Intent Manager and a Remote Procedure Call (RPC) Manager. Within an application environment or "framework" of a search application, the computing device performs operations under the direction of the Intent Manager and the RPC Manager.

When a user initiates a search in a search application, the mobile device receives user input, for example, by means of either an input/output device or an intelligent virtual assistant. Regardless of the mode, the search application registers the user input as textual data ("query text"). The query text may include a prefix of a word or an incomplete phrase.

In a first step, after registering the query text, the RPC Manager accesses an on-device personal query set (e.g. personal dictionary) having cached query texts ("personal queries") of the

user. The RPC Manager then extracts personal queries that are similar to the query text. Determination of similarity may involve, for example, N-gram probability calculations and/or other natural language processing techniques that may indicate a high likelihood of word or phrase similarity. In a second step, the RPC Manager presents at least one similar personal query ("query suggestion") to the user as a selectable graphical element displayed on the input/output device. These first two steps execute sequentially, and together constitute a single RPC referred to as a query completion RPC. In a third step, the RPC Manager searches for content using the query text as the search criteria and returns search results. This third step is referred to as an answer suggestion RPC. The RPC Manager executes the query completion RPC and the answer suggestion RPC in parallel.

Example operations of the RPC Manager are illustrated in Figure 2. While a user is typing in a search application, the RPC Manager executes the query completion RPC. Operations of the query completion RPC include identifying similar personal queries to the query text ("Wea") and presenting a similar query ("Weather San Jose") as a selectable query suggestion. The RPC Manager also executes the answer suggestion RPC in parallel. Operations of the answer suggestion RPC include searching for content using the query text ("Wea") as the search criteria. If the answer suggestion RPC discovers content, it returns the search results, and the search application presents the content to the user. As illustrated in Figure 2, the answer suggestion RPC did not discover content using the query text "Wea" as the search criteria and, therefore, the search application did not present content to the user. When the user selects the query suggestion "Weather San Jose," the query text is updated and the RPC Manager executes the answer suggestion RPC again. The answer suggestion RPC uses the updated query text ("Weather San

Jose") as the search criteria, resulting in the search application presenting search results for weather in San Jose, California.

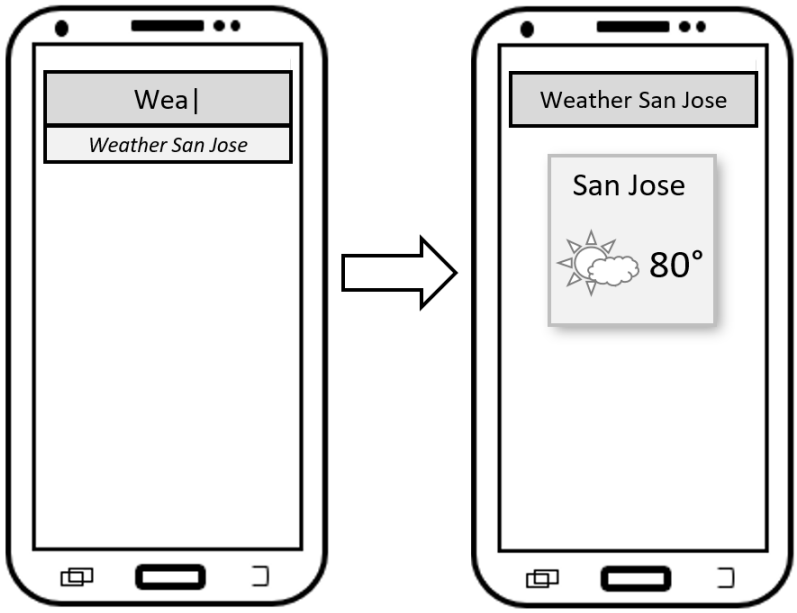

Figure 2

As illustrated in Figure 2, the search application utilizes an answer card user-interface (UI) and presents an answer to a search in an answer card. The answer card is a rich user-interface (UI) selectable graphical element containing information concerning weather in San Jose, California. While answer cards may include a brief textual description or summary of content on corresponding pages, the mobile device requires further user interaction (e.g., selecting the answer card, instructing an intelligent virtual assistant) before navigating to content linked to the search results.

In a fourth step, the Intent Manager associates a query suggestion to a corresponding answer card for a given query text. The association may involve the Intent Manager utilizing an algorithmic (e.g., machine-learned) technique configured to identify that the submission of a query suggestion results in a search application presenting one or more corresponding answer cards. In addition, association may include the production of metadata relating the query text to the

submitted query suggestion. The Intent Manager can then cache the metadata on-device. For instance, referring back to the example illustrated in Figure 2, when the user typed the query text "Wea," the search application, by means of the query completion RPC, presented the query suggestion ("Weather San Jose"). Once the user submitted the query suggestion (e.g., by clicking the query suggestion), the answer suggestion RPC presented an answer card. As a result, the Intent Manager associates the submitted query suggestion ("Weather San Jose") to the corresponding answer card for the given query text ("Wea"). The association involves the Intent Manager producing metadata relating the query text ("Wea") to the submitted query suggestion ("Weather San Jose"). Next, the Intent Manager caches the metadata on-device. The Intent Manager can routinely update or synchronize the cached metadata to verify association is still desirable.

In a fifth step, the Intent Manager searches through the cached metadata for a query text. Upon finding the query text, the Intent Manager extracts the related query suggestion. For example, the Intent Manager may search for the query text "Wea" in the cached metadata. Upon finding the query text in the cached metadata, the Intent Manager extracts the related query suggestion ("Weather San Jose"). After extracting the related query suggestion, the Intent Manager directs the RPC Manager to issue an answer suggestion RPC using the related query suggestion as the search criteria.

Figure 3, below, illustrates an example situation in which a user, having previously searched for the weather in San Jose, California (e.g., as illustrated in Figure 2), is presented with a query suggestion and an answer card.

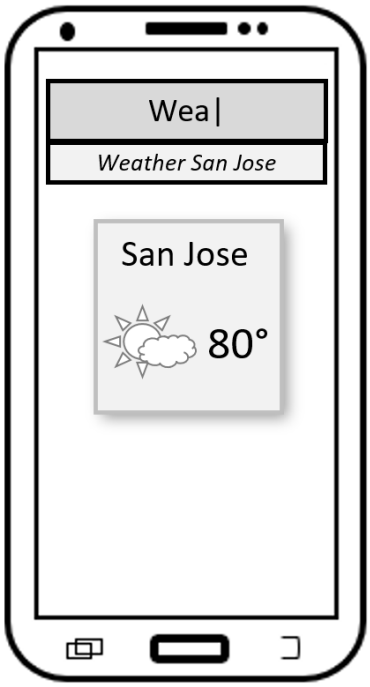

Figure 3

As illustrated, the user types a query text (“Wea”) and the search application presents both a query suggestion (“Weather San Jose”) and an answer card. The search application presents the query suggestion because the RPC Manager issues a query completion RPC and discovers a personal query similar to the query text (“Wea”). The RPC Manager also issues a first answer suggestion RPC in parallel by using the query text (“Wea”) as the search criteria.

The search application presents the answer card because the Intent Manager searches through cached metadata, discovers an identical query text (“Wea”), and extracts the related query suggestion (“Weather San Jose”). As a result, the Intent Manager directs the RPC Manager to issue a second answer suggestion RPC using “Weather San Jose” as the search criteria. The RPC Manager issues the second answer suggestion RPC in parallel with the first answer suggestion RPC. In a final step, the Intent Manager compares the related query suggestion to the query

suggestion returned by the query completion system. The Intent Manager determines that the query suggestions are identical and, as a result, cancels the first answer suggestion system RPC.

Throughout this disclosure, examples are described where a mobile device may analyze text associated with a user, for example, the query text the user types into a search application on the mobile device. 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, and/or features described herein may enable collection of information (*e.g.*, a user's preferences), and if the user is sent content or communications from a server. The mobile device can be configured to only use the information after the mobile device receives explicit permission from the user of the mobile device to use the data. For example, in situations where the mobile device caches personal queries, individual users may be provided with an opportunity to provide input to control whether programs or features of the mobile device can collect and make use of the data. Further, individual users may have constant control over what programs or applications can or cannot do with the information. In addition, information collected may be pre-treated in one or more ways before it is transferred, stored, or otherwise used, so that personally-identifiable information is removed. For example, cached metadata of a user may be treated so that no personally identifiable information can be determined for the user. For example, a user's geographic location contained in cached metadata may be generalized where location information is obtained (for example, 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 whether information is collected about the user and the user's device, and how such information, if collected, may be used by the mobile device and/or a remote computing system.

References:

[1] Patent Publication: US20170004210A1. Selecting and Presenting Content Relevant to User Input. Priority Date: November 4, 2009.

[2] Patent Publication: US20160371263A1. Apparatus, System and Method for a Semantic Editor and Search Engine. Priority Date: July 8, 2013.

[3] Patent Publication: US20170097986A1. Methods, Systems and Techniques for Providing Search Query Suggestions based on Non-Personal Data and User Personal Data According to Availability of User Personal Data. Priority Date: October 5, 2015.