

# Technical Disclosure Commons

---

Defensive Publications Series

---

July 2023

## Utilizing a Human Computer Interaction Technique for Enabling Non-Disruptive Exploration of App Contents and Capabilities in a Query Recommendation System

Keun Soo Yim

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Yim, Keun Soo, "Utilizing a Human Computer Interaction Technique for Enabling Non-Disruptive Exploration of App Contents and Capabilities in a Query Recommendation System", Technical Disclosure Commons, (July 31, 2023)

[https://www.tdcommons.org/dpubs\\_series/6092](https://www.tdcommons.org/dpubs_series/6092)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## **Utilizing a Human Computer Interaction Technique for Enabling Non-Disruptive Exploration of App Contents and Capabilities in a Query Recommendation System**

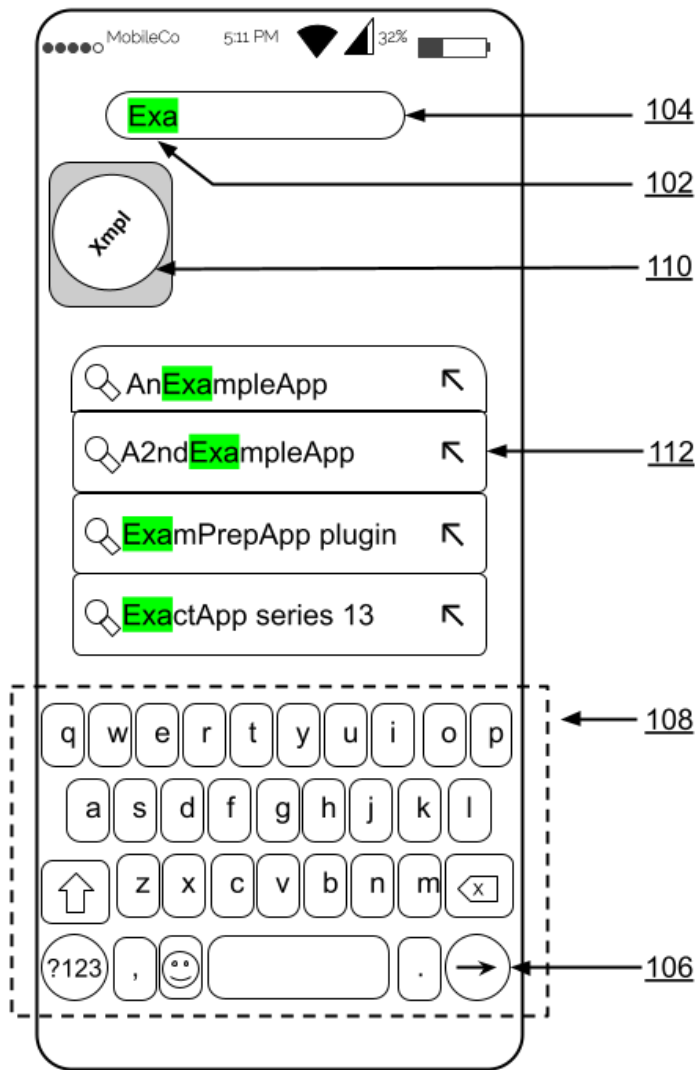
### **ABSTRACT**

Conventional techniques for launching apps do not provide any facility to quickly launch app contents or app capabilities. This disclosure describes techniques for quickly launching app capabilities and contents, surfacing those capabilities/contents that result in high user interaction rate (UIR) without harming a total clicks metric. In contrast to conventional techniques, app contents and capabilities with high potential UIR are adaptively determined using heuristics and user-permitted interaction data. A quick scroll button advantageously enables providing a scroll interface with suggestions that are otherwise hidden behind a virtual keyboard or not displayed. App contents and capabilities with high potential UIR are determined with low computational and UI costs. By directly enabling the scrolling and selection of relevant, popular, or personalized app content and capabilities, the user interface provides enhanced convenience and speed of operation.

### **KEYWORDS**

- Quick launch
- Deep link
- Query recommendation
- User interaction rate (UIR)
- App capability
- App content
- Total clicks metric
- User experience
- Reinforcement learning alternative

BACKGROUND



**Fig. 1: Query recommendation displaying matching apps**

As illustrated in Fig. 1, on smartphones and other devices, a user can quickly launch an app by typing the prefix (starting few characters, 102) of the app name in a query box (104) and pressing a quick launch button (106) on a virtual keyboard (108). The quick launch button is typically situated at the location of the enter button, and the enter and the quick launch button can contextually switch between each other. The quick-launch button is indicated by a right-pointing arrow (106). The virtual keyboard is rendered in the bottom portion, e.g., the bottom

third of the device screen. An app that has a name that matches or nearly matches the prefix is highlighted (110). Other possible completions are listed (112). In this manner, the highlighted app and the possible completions are boosted, e.g., shown on the screen against competing content.

Upon the user tapping the quick launch button, the highlighted app launches without the user having to tap on the app icon (110), which can be relatively far away from the virtual keyboard, e.g., towards the top of the screen as shown in Fig. 1. The app of interest can be launched with just a keyboard swipe or two, possibly with just one hand. Fig. 1 is an example of a query recommender, a module that accepts the prefix of an app and produces possible and relevant app completions that enable rendering of the user interface.

An app can be described by its capabilities and its contents. App capabilities include app shortcuts and app actions. For example, for a clock app, app shortcuts can include the creation of an alarm or timer, the creation of start or stop buttons of a stopwatch, etc. App contents can include:

- **App's top suggestions:** a list of top app content suggestions per app.
- **App's category suggestions:** a list of app content suggestions per app per category. For example, for the clock app, one category can be alarms and another can be timers.
- **App's entity suggestions:** a list of app content suggestions per app per specific matching entity (e.g., the 5 AM alarm of the clock app, the 30-minute timer for the clock app, etc.).

Current techniques do not include features that enable a user to quickly launch app contents or app capabilities. This presents a hurdle in bringing users quickly to relevant activities or pages in the target app, since the user needs to navigate the app user interface to get the app content or capability of interest. User experience design (UXD) and technical challenges that

hinder the launching of app contents and capabilities identified by a query recommendation system include:

- An app can have many capabilities and contents associated with it. For example, a clock app can include multiple timers, a stopwatch, and multiple alarms in addition to basic time display. If the user types in ‘clo’ (the starting few characters of the app name), the user input text does not provide sufficient information to enable determination of the app capabilities that are to be boosted: clock→displayTime, clock→alarmView, clock→alarmOpen, clock→5A (for a previously set 5 AM alarm) are all reasonable possibilities to be surfaced. On a typical smartphone, the size limitations of the screen make it difficult to show all, or even a select few, of the many capabilities and contents of various apps. Indeed, it may be unclear which of the contents and capabilities of even a popular app are to be boosted.
- Boosting (displaying a particular option or action) of an app comes at an opportunity cost of hiding another app or another action of the same app with a higher user interaction rate (UIR).
- An unboosted app (an app content/capability that is not surfaced on the screen, e.g., hidden behind or below the keyboard, or not surfaced at all) poses a discoverability problem since users are not made aware of the existence of the app or app content/capability.

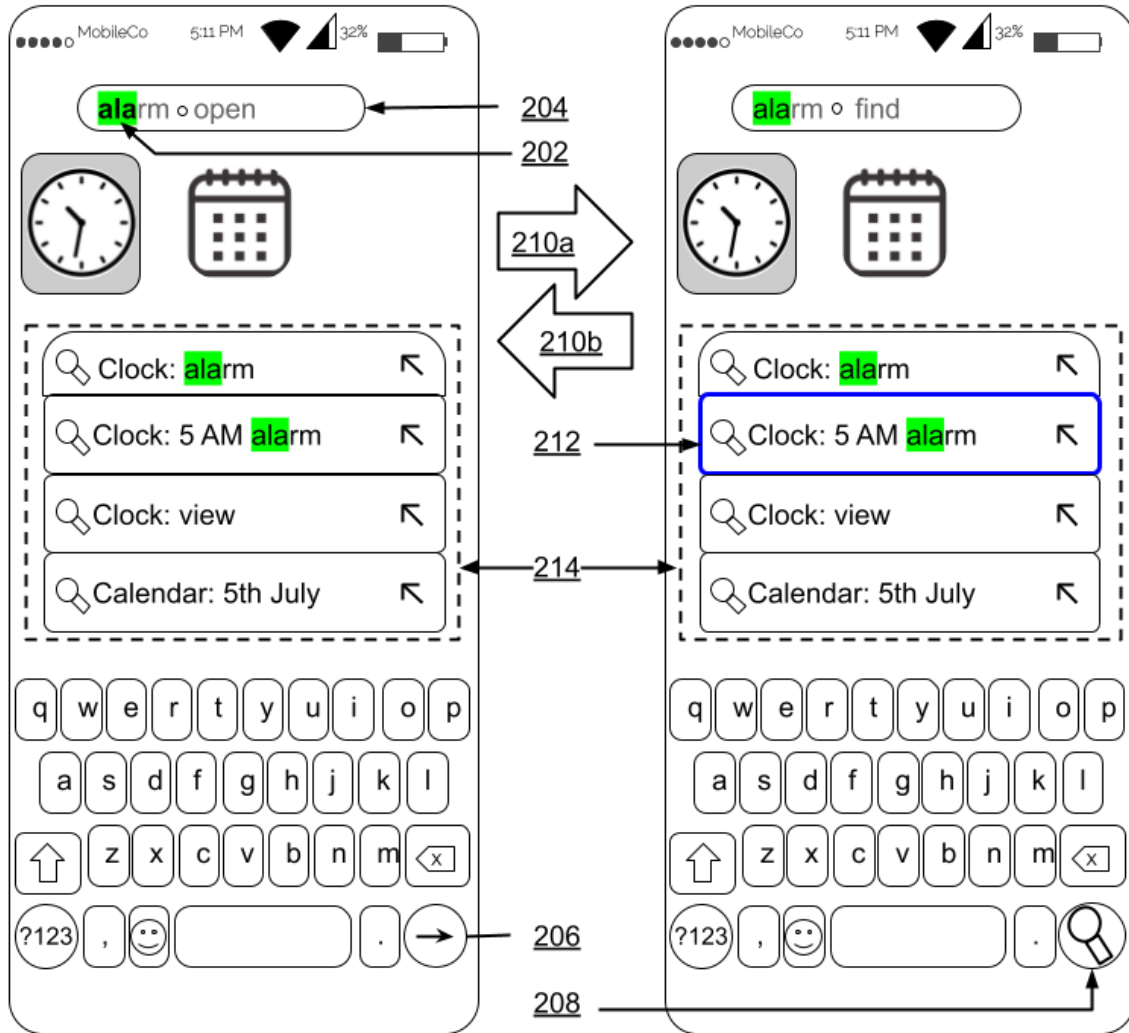
The total clicks metric is defined as the sum of user interactions on the on-device app suggestions (including suggestion to open an app, app content, app capabilities, etc.) as well as web suggestions (e.g., a web-based resource that matches partial query and can be indicated in the interface). Here, interactions include clicks and impressions that last greater than a threshold

amount of time. The total clicks metric is closely related to user interaction rate (UIR), which is the number of times a user clicks on (or otherwise selects) a suggestion divided by the number of times the suggestion is shown.

Once UIR data is obtained, it is possible for app contents and capabilities with total clicks metric higher than web suggestions (when the quick launch interface supports both types of suggestions) to be boosted. However, determination of app contents/capabilities with high total clicks metric can be costly. For example, making such a determination using machine learning (e.g., reinforcement learning-style exploration) techniques is costly for on-device query recommendations, given the tight fetching latency and system health requirements of on-device platforms and apps. Besides, reinforcement learning, which involves stages of exploration and exploitation, can harm total clicks and UIR metrics during the exploration stage to a certain degree.

## DESCRIPTION

This disclosure describes techniques to quickly launch apps, including app capabilities and contents, surfacing app capabilities/contents that likely result in high user interaction rate without degrading the total clicks. Suggestions for app contents and app capabilities are enabled, while all or most are not initially boosted. In contrast to conventional reinforcement learning based techniques, app contents and capabilities with high potential UIR are adaptively determined using heuristics and user experience techniques. Avoiding the exploration stage of reinforcement learning allows determination of high-UIR suggestions in a manner neutral for the total clicks metric.



**Fig. 2: Quick launcher displaying app content or capabilities**

Fig. 2 illustrates an example of quick launching app content or capabilities. When the prefix typed by the user ‘ala’ (highlighted in green, 202) matches the content or capability of an app, without necessarily matching the name of an app itself, the prefix is completed (‘alarm’) to the suggestion name shown in the text box (204). The auto-completed letters of the suggestion can be rendered in lighter font.

The quick launch button (206), located at position of the enter key of the virtual keyboard, transforms (210a) to a quick-scroll button (208), represented by a dedicated icon (e.g.,

a magnifying glass), enabling the user to scroll and see, in ascending order of rank, the relevant app content or capability suggestions (214), which may otherwise be hidden behind the keyboard. Auto-completed command text in the text box (204) can change (e.g., from ‘open,’ indicating quick launch, to ‘find,’ ‘search,’ or ‘scroll’ indicating quick scroll) to indicate the mode change from quick launch to quick scroll.

Some examples of user-typed prefixes triggering suggestions for app content or capabilities are described below. A prefix ‘ala’ can be completed to ‘alarm view’ or ‘alarm open’ of a clock app, although the prefix ‘ala’ itself doesn’t include any string from the app name ‘Clock.’ A prefix ‘5’ can be completed to a ‘5 AM alarm’ of a clock app or a ‘5th July appointment’ of a calendar app, although the prefix ‘5’ appears in neither of the app names (‘Clock’ or ‘Calendar’). The name of a movie can trigger a suggestion of a streaming app with a shortcut to the movie. Partially typing a destination (‘5th Stre’) can bring up a ride-hailing app.

In general, once UIR data is obtained, app contents and capabilities with total clicks metric higher than web suggestions are boosted. This is analogous to the exploitation stage of reinforcement learning, without the costs associated with reinforcement learning. However, if the user-typed prefix directly matches the name of an app, rather than app content or capability, the matching app is given priority. In this case, the quick-launch button does not transform to a quick scroll button, and instead quick launch suggestions with the matching app names are displayed. Upon the user tapping the quick launch button, the selected app is launched. As explained earlier, if the user-typed prefix does not match the name of an app, then app capabilities or app contents that can result in high UIR are surfaced.

The quick scroll button advantageously enables suggestions that would otherwise be hidden behind the keyboard or not displayed are provided in a convenient scrolling interface.



The present position in the scrolling interface can be indicated suitably, e.g., by a blue rectangle (212). When the user taps on the quick scroll button, the quick scroll button is transformed back (210b) to the original quick launch button, indicating that the selected app will be opened at its relevant section. When the user taps on the original quick launch button, the target app is opened at the selected app content or capability suggestion (e.g., '5AM alarm').

If the match is with the category preview (e.g., 'alarm'), the app category suggestion block which can contain multiple entity suggestions in the category, is scrolled down to. A specific entity suggestion ('5 AM alarm') within the app category suggestion block (5 AM alarm, 7 AM alarm, 9 AM alarm, etc.) can be selected by the user. If the match is with the entity preview (e.g., 'timer'), the specific entity suggestion of the app is scrolled down to. In addition to showing the completed suggestion name, the suggestion type (e.g., app shortcut, app content, content type, app name, etc.) is shown. As mentioned earlier, the matched suggestion can be highlighted, e.g., using a blue rectangular box. Alternative to scrolling, the first matched app content or capability suggestion can be directly opened using the quick launch button. In another alternative, the quick scroll button can be updated to a quick launch button when the scroll button is tapped.

In this manner, the user can discover app content or capability suggestions that were originally unboosted in a query recommendation interface. Even without boosting, the UIR and the total clicks metric of the app content or capability can be determined while the query intent is clear and thus captured by the described techniques. Conventionally, quick launching typically covers queries with explicit mention of app names. The described techniques augment conventional launching by triggering for queries that mention app contents or capabilities. With sufficient user interaction, an app content or capability can be boosted and shown on the screen

right away, as follows. Upon prefix-matching, the quick launch button is shown for boosted app content or capability without the possibility of transformation to quick scrolling, thereby enabling opening of the app content or capability in the target app without intermediate scrolling. Direct boosting of an app content or capability after sufficient user interaction is approximately equivalent to the exploitation stage of reinforcement learning. However, such direct boosting is achieved without the high computational cost of reinforcement learning and without the damage to the total clicks metric that may be caused by the exploration stage of reinforcement learning.

The act of a user glancing on a boosted suggestion for a sufficiently long time (as is typical for weather reports, stock prices, etc.) and closing the query recommender (without tapping on the boosted suggestion) may be considered a case of good abandonment. Such an event may be counted such that the boosted suggestion is counted towards the total clicks. Lacking a tap or a sufficiently long glance, the suggestion can be de-boosted since its contribution to the total clicks is smaller than initially measured. In this manner, the described dynamic state tracking can enable the discovery and display of suggestions with high total clicks metric.

In this manner, app contents and/or app capabilities with high potential user interaction rate are determined with a low computational and UI costs that can be supported on consumer-grade, relatively small form factor mobile devices. The techniques enable an improved quick launch interface with minimal to no harm to the total clicks metric. By directly enabling the scrolling and selection of relevant and popular app content and capability, the techniques offer enhanced convenience and speed of operation.

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 interaction with a user interface including selections made by the user and/or gaze data, a user's queries, apps on a user device, app contents, 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. 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 for quickly launching app capabilities and contents, surfacing those capabilities/contents that result in high user interaction rate (UIR) without harming a total clicks metric. In contrast to conventional techniques, app contents and capabilities with high potential UIR are adaptively determined using heuristics and user-permitted interaction data. A quick scroll button advantageously enables providing a scroll interface with suggestions that are otherwise hidden behind a virtual keyboard or not displayed. App contents and capabilities with high potential UIR are determined with low computational and UI costs. By directly enabling the scrolling and selection of relevant, popular, or personalized app content and capabilities, the user interface provides enhanced convenience and speed of operation.

## REFERENCES

[1] Abner Li, "Pixel Launcher, letting you quick launch apps from search, rolling out," available online at <https://9to5google.com/2023/05/16/pixel-launcher-quick-launch/#:~:text=Now%20it%27s%20rolling%20out%20and,keyboard%20will%20open%20it%20immediately> accessed on Jul. 16, 2023.