

Technical Disclosure Commons

Defensive Publications Series

March 13, 2019

Voice search capabilities for lists and notes

Amitabh Saikia

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Saikia, Amitabh, "Voice search capabilities for lists and notes", Technical Disclosure Commons, (March 13, 2019)
https://www.tdcommons.org/dpubs_series/2018



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.

Voice search capabilities for lists and notes

ABSTRACT

Different types of user interfaces are available on various computing devices to access lists and notes, e.g., a graphical user interface, a voice user interface, etc. A search engine stack is utilized to process user requests, e.g., voice queries, which enables available matches to be ranked and selected. Use of the search operation provides a superior user experience when compared to simple relational database searches. Integration of the search engine stack enables fuzzy matching of user requests to available notes and lists. If the user provides permission, past interactions are utilized. Interpretation or corrective action regarding a query, based on past interactions, is performed only upon permission from the user.

KEYWORDS

- Virtual assistant
- Personal digital assistant
- Voice UI
- Note taking
- Home speaker
- Fuzzy search
- Search engine

BACKGROUND

Various computing devices such as mobile phones, laptops, tablets, wearable devices, home speakers, smart appliances, etc. provide interfaces for users to maintain lists or notes. For example, a virtual assistant or personal digital assistant, can provide list and note taking

features. Users prefer to maintain a single set of lists and notes that is synchronized across and easily accessible to the user from any device.

DESCRIPTION

This disclosure describes techniques to access notes and lists on computing devices with different types of available user interfaces. Per techniques of this disclosure, a search operation is integrated into operational paths for user access and synchronization for lists or notes. The search operation uses a search engine stack to process user requests (e.g., voice queries) and enables available matches to be ranked and selected. Use of the search operation provides a superior user experience as compared to simple relational database searches.

Integration of the search operation enables fuzzy matching of user requests and can be particularly useful in scenarios where user access is through a voice-based interface (via a virtual assistant, for example). For example, a user that wishes to add an item to a shopping list titled ‘XYZ shopping list’ can specify the term 'shopping' and a list that matches the term most closely can be retrieved automatically to add the item.

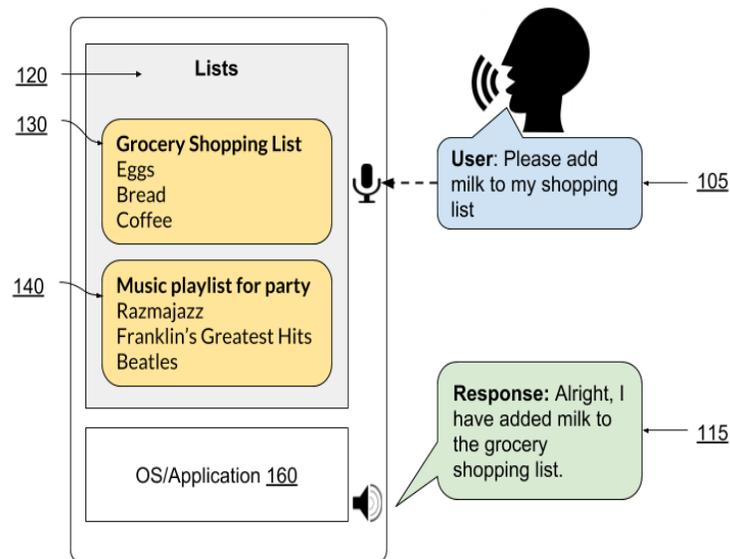


Fig. 1: A fuzzy search enables user access to notes/lists via a voice based interface

Fig. 1 illustrates an example of a voice-interface based access of lists (120) by a user. In the illustration, the lists are depicted as stored on the user device; however, lists can also be stored at a central server, e.g., a cloud-based server, etc., and OS/Application (160) be configured to operate in conjunction with the central server. In the illustrated example, the user has two lists - a grocery shopping list (130) and a music playlist (140). The user makes a request (105) via a voice query, e.g., to a virtual assistant: “Please add milk to my shopping list.”

Per techniques of the disclosure, a search operation utilizes the terms “shopping list” from the user query. The search operation leads to a determination that the user intended to update the grocery shopping list. For example, such determination is based on the term “shopping list” matching “grocery shopping list” and not matching “music playlist”; based on the query parameter “milk” matching the phrase “grocery” or existing contents of the list “Eggs, bread, coffee,” etc. The identified list is accordingly updated and a confirmation message (115) is provided to the user. Users are enabled to access and update lists and notes via a voice-based user interface (VUI) and/or a graphical user interface (GUI), using any of the types of devices listed above, based on available input mechanisms.

Different types of user actions on notes and lists that can be performed include, e.g., creation of a new note or list, locating a note or list, updating or augmenting a note or list, marking/unmarking a note/list (e.g., marking a listed task as completed), deletion of a note or list, sharing or unsharing with another user, moving a note to a list, addition or deletion of an item, updation of an item, moving or copying an item from one note or list to another, reordering items within a note or list, etc. Providers can have different capabilities, and the user interface is suitably tailored to match user actions to provider capability.

When updating a note, prior context may be utilized. For example, if the list already has the item “milk” (e.g., added by a previous command), a user command such as “change that to soy” is interpreted to update the list to replace milk with soy milk. A user command such as “mark it done” or “delete that” causes the item mentioned in the immediately preceding command to be marked as completed. When lists or notes are maintained on a server, the server may enable such operations to be performed via an API, e.g., a user command is sent to the server as an API call, including user-specified parameters such as list names, item descriptions, operation to be performed, etc.

In the case of user access via a GUI, an identifier that is stored (and tracked) for each note or list enables direct user operation on a note or list. In the case of a voice-based user interface, access to the last created or accessed list or note is provided. Access to other lists and/or notes via a voice-based UI is enabled through the fuzzy search operation as described earlier. In some implementations, annotation attachments are utilized to augment information stored in a note. Not all actions may be supported by either the graphical UI or the voice UI.

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, 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

Different types of user interfaces are available on various computing devices to access lists and notes, e.g., a graphical user interface, a voice user interface, etc. A search engine stack is utilized to process user requests, e.g., voice queries, which enables available matches to be ranked and selected. Use of the search operation provides a superior user experience when compared to simple relational database searches. Integration of the search engine stack enables fuzzy matching of user requests to available notes and lists. If the user provides permission, past interactions are utilized. Interpretation or corrective action regarding a query, based on past interactions, is performed only upon permission from the user.