November 2020

Automatically Generating Variant Queries of an Input Query Based on Input Mode

Matthew Sharifi

Victor Carbune

Automatically Generating Variant Queries of an Input Query Based on Input Mode

ABSTRACT

This disclosure describes techniques for translating an input query provided to a digital assistant to equivalent queries having a different input mode than the input query, such as typed or spoken input mode. A query provided by the user in one input mode is translated to variants for one or more other input modes based on grammars, user-permitted query logs, and/or machine learning models. Components of the digital assistant are updated with both the query and the equivalent query. The described techniques can be used to adapt digital assistant speech recognition and query response generation based on previous user queries, provide a complete query history, and provide improved auto-completion suggestions. Such features improve the experience of using a digital assistant for users that switch between different input modes in various interactions.

KEYWORDS

- Digital assistant
- Virtual assistant
- Input mode
- Spoken query
- Typed query
- Query history
- Query variant
- Query translation
- Query transformation
- Query embedding

BACKGROUND

Users frequently interact with voice-enabled digital assistants available via smart devices such as phones, watches, smart speakers, smart displays, etc. A digital assistant performs various tasks for users and finds answers to questions that the user inputs to the digital assistant, through

natural and conversational interactions. Users often input queries to the digital assistant via speech, but may also provide input by typing queries via a keyboard, a touchscreen, or other input device.

While spoken queries or commands are often convenient and enable interactions with the virtual assistant via computing devices in scenarios where other forms of input are inconvenient or are not possible, it is not always optimal. For example, in a noisy environment, it might be difficult for a user's speech to be understood by the digital assistant. In such a context, the user is better off typing a query instead, and may choose that option. Also, in a public space, a user might not feel comfortable to speak their query out loud since others nearby might overhear their request and/or the digital assistant response.

As a result, users may switch between typed and spoken queries to a digital assistant, choosing the mode based on their current context, the type of query, or other factors. Spoken and typed queries are typically phrased differently. For example, spoken queries are typically more conversational, while typed queries are shorter and more keyword-based. Such differences in the forms of the queries means that techniques such as adapting of query recognition and response generation based on previous interactions or query history (if permitted by the user) may not work effectively across different input modes, e.g., because the two forms of queries may not be seen as equivalent or related.

DESCRIPTION

This disclosure describes techniques to generate variant queries for an input query provided via a first input mode such as typing, where the variant queries are related to a different input mode, such as speech. For an input query provided via a particular input mode, equivalent queries are generated for other input modes and used to update the query history (e.g., by storing

the equivalent queries in the user's query history for the other modes) and/or to update machine learning models used by the digital assistant. Such generation and use of variant queries allows digital assistant responses to future queries from the user to be more satisfactory regardless of the input mode in which the user inputs their queries.

For example, the described techniques transform between typed and conversational versions of a query. Users often switch between different input modes (speech vs. typing) depending on their current situation. Queries are phrased differently across these modes. For example, when providing a spoken query, a user might ask "how old is Barack Obama?" in a conversational format, whereas when typing a query, they might instead input "Barack Obama age" in a more keyword-focused format.

By automatically mapping such queries to one another, improved cross-mode adaptation, query history, and other digital assistant performance and features can be achieved. For example, users can be presented with query histories and auto-completion suggestions to the beginnings of queries that take into account queries from any input mode. Also, the equivalent query that has been transformed from an input query can be output to a user. This allows users to better understand how their query was interpreted.

The described techniques are implemented upon specific user permission to access a user's data, e.g., speech input from the user, history of user interactions with a digital assistant (including previous queries and received responses), user context information, etc. Users are provided with options to grant permissions to and/or to disable features entirely. The user can enable or disable techniques discussed herein for particular locations, time periods, or for other conditions.

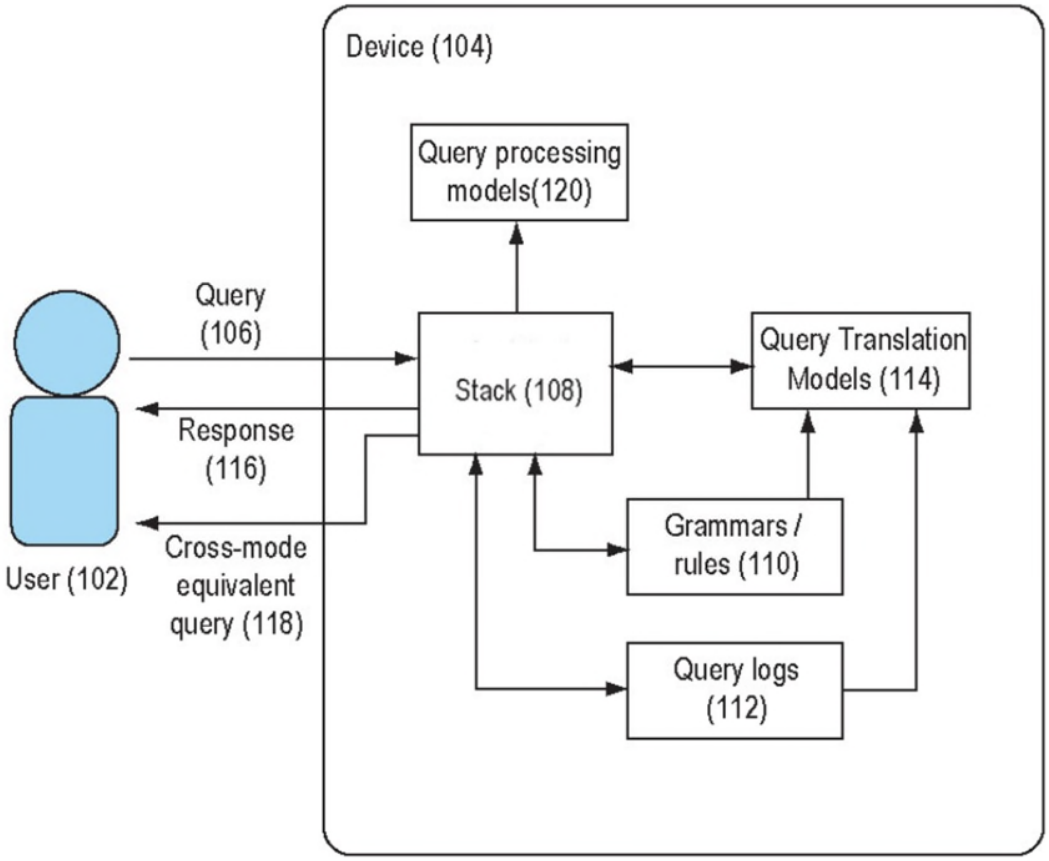

Fig. 1: Translating an input query to a query of a different input mode

Fig. 1 shows an example of operational implementation of techniques described in this disclosure. A user (102) activates a digital assistant available via a device (104) by a standard method, e.g., a voice command (hotword), input gesture, press of a button, etc. The user chooses a mode of input for a query (106), e.g., to type the query or speak the query. This may depend on the current context of the user, e.g., convenience of one input mode over the other, noisy environment, etc. The query can be a question or request seeking an answer or other information about facts, user schedules, device characteristics, or other subject.

The query is provided as input to a stack (108) that processes the query. For example, the stack can include an Automatic Speech Recognition (ASR) system that includes trained machine

learning models and/or other components to perform speech recognition. The stack can include natural-language understanding (NLU) components to process the query. Via query processing, the query is recognized and parsed and a response to the query is generated, e.g., based on data sources such as a structured knowledge graph, web pages, etc.

At the same time as the query is processed to produce a response by the stack, query translation is performed, from the input mode of the query (e.g., typed) to other modes of input (e.g., speech), to generate a cross-mode equivalent query (or variants thereof). For example, if the typed query is "Barack Obama age," the variants can include "what is Barack Obama's age?" "how old is Barack Obama?" "When was Barack Obama born?" etc. Thus, a typed query is automatically translated to one or more spoken query variants. Transformations can be generated between any pair of query input modes.

In some cases, the query translation into the target input mode is performed only if there is historical data showing that the user has used the other input mode at some point in the past. For example, a user that has never typed a query to the digital assistant is not likely to benefit from translation of the query into a typed mode since the digital assistant is likely to only process queries having the spoken input mode for this user.

Query translation can be grammar or rule-based, e.g., based on query parsing as controlled by grammars/ rules (110). Based on parsed portions of the query, a cross-mode equivalent of the input query is constructed, e.g., a typed query is constructed if the query is spoken, or a conversational or spoken query is constructed if the query is typed. For example, grammars can instruct to generate a cross-mode equivalent query that has a particular sequence of types of words (verbs, nouns, modifiers, etc.).

If permitted by the user, query translation can also be based on previous queries input by the user as accessed in a query history including query logs (112) which store queries previously input by the user and the responses provided to those queries. A cross-mode query is determined based on previous queries of the same input mode that caused responses that are the same or similar to the response to the current input query. For example, if the current query is spoken, the query logs are accessed to determine which previous typed queries have achieved the same response that has been generated for the current query, and those typed queries are used as the basis for generating a cross-mode equivalent query.

Query translation models (114), once trained, can be used to translate the input query to a cross-mode equivalent query. The models are trained based on the data and results of generating equivalent queries as described above, utilized with specific user permission. For example, the query translation models can include a query embedding model. This model can be a neural network model that embeds spoken or typed queries in the same space. The model can be trained on the query logs. At query time (after training), the cross-mode equivalent query to the input user query is determined by embedding the input query and looking for the nearest neighbor in the embedding space that corresponds to the other input mode(s). Alternatively, the query translation models can include a dedicated query translation model that is trained on the same data described above. This can be a standard sequence-to-sequence model, such as a transformer or recurrent neural network.

A response (116) to the user's input query is output to the user as it typically would be output based on the stack processing of the input query. For example, the response can be output as speech by a text-to-speech engine or other component. In some cases, the response can be

additionally or alternatively output in visual form, e.g., displayed as text on a screen or other display device.

In addition, the cross-mode equivalent query (118) that is generated as described above can also be output to the user, which may clarify to the user how the input query was interpreted by the digital assistant. For example, if the input query was typed, a conversational (speech-based) version of the query can be output as visual text or as speech.

After the response is output, various components of the digital assistant are updated based on the input query, the cross-mode equivalent query, and the response. The update of components is performed twice: once based on the original input query, and once based on the equivalent query. If multiple equivalent queries are identified, an update is performed for each variant. For example, a query history including the query logs are updated such that both forms of query are associated with the response. Query processing models (120), such as machine learning models and other user-based machine learning models are also updated. For example, biasing models are used to adjust and adapt recognition of queries and/or weight the generation of responses to the queries, based on previous interactions with the user (e.g., as indicated in the query history and other data).

The updated models can then be applied for future queries input by the user. For example, this allows a spoken query, “how tall is Barack Obama?” to be represented as the equivalent typed query “Barack Obama height” in their query history when examining queries in a typed input mode. The equivalent query can be displayed to the user as an auto-completion suggestion if the user retypes the same query or a similar query at a future time.

In addition to the above, described techniques may be used to help users expand on queries that are not fully specified. For example, if a user types the query “pizza,” multiple

variants of an equivalent conversational (spoken) query may be determined, such as “how to cook a pizza” and “where to order a pizza.” In this situation, the user can be provided with an option to choose one of the interpretations, and provide the corresponding results.

The described techniques can be implemented in various configurations as permitted by the user. The digital assistant can be provided via a device, e.g., a user device (such as a smartphone, wearable device, etc.), a smart speaker, or other device. For example, the functionality of the digital assistant and/or other components may be incorporated within one or more applications or in a device operating system. In another example, if the user permits, one or more of the digital assistant, models, text-to-speech engine, data source, and/or other components can be executed on a server that is remote from the user device.

The various features of the system are implemented only with user permission to access user data that serves as input to the system. 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., context information about a user, a user’s speech input and commands, query history, responses to prior queries, a user’s preferences, a user’s location, a user’s calendar items, etc.), and if the user is sent content or communications from a server. Certain techniques are not implemented if users deny permission. 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. 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 translating an input query provided to a digital assistant to equivalent queries having a different input mode than the input query, such as typed or spoken input mode. A query provided by the user in one input mode is translated to variants for one or more other input modes based on grammars, user-permitted query logs, and/or machine learning models. Components of the digital assistant are updated with both the query and the equivalent query. The described techniques can be used to adapt digital assistant speech recognition and query response generation based on previous user queries, provide a complete query history, and provide improved auto-completion suggestions. Such features improve the experience of using a digital assistant for users that switch between different input modes in various interactions.