October 2020

INPUT METHOD EDITOR (IME) SUPPORTING MULTIPLE HYPOTHESES APPROACH

Victor Carbune

Matthew Sharifi

INPUT METHOD EDITOR (IME) SUPPORTING MULTIPLE HYPOTHESES APPROACH

ABSTRACT

A system is described that enables an input method editor (IME) on a computing device (e.g., a wearable device or a mobile device) that provides multiple hypotheses. There are various different mechanisms for inputting a user's input on a device, such as touch and swipe typing, voice input, handwriting recognition, camera input, optical character recognition (OCR), etc. The device may enable an IME to process a user's input and return, to an application, multiple hypotheses in a standardized machine-readable form based on the input. The application may select one of the returned hypotheses to use as final input. By enabling the IME to return multiple hypotheses based on a user's input, the device enables the application to use multiple hypotheses to provide an improved (e.g., more accurate) result.

DESCRIPTION

There are a number of different mechanisms for composing and inputting text on mobile devices such as touch and swipe typing, voice input, handwriting recognition, optical character recognition (OCR), etc. Users may wish to use these different mechanisms across any apps on the device, such as to compose a short message in an instant messaging application, to search for a product name on a shopping application, or find a particular movie on a video streaming application. In order to support various input mechanisms, input method editors (IMEs) may be implemented to transform a received user input to a standardized machine-readable form, such as text. One major challenge is that there are very different levels of reliability for each of the input mechanisms. For example, voice input and handwriting recognition may have higher word error rates than typing. Such varying levels of reliability may produce results of varying quality. As such, it may be desirable to improve the quality of user input results.

Techniques are described that enable a device to provide multiple hypotheses to an application. For instance, as opposed to providing a single hypothesis based on user input, an IME may provide an application with multiple hypotheses (e.g., multiple candidates) based on the user input. The application may then select one of the hypotheses (e.g., based on context or other information available to the application, but that may not necessarily be available to the IME). The device may include one or more sensors that collect user input data. The collected data may be stored in a memory or other storage device of the device. The device may utilize input method editors (IMEs) to process the user input data to generate multiple hypotheses. In operation, the device may determine a confidence score (e.g., using one or more machine training models) for each of the multiple hypotheses and select the top N hypotheses according to their confidence score. The device may further provide the top N hypotheses to an application for selection.

In some examples, the application may return an indication of the selection to the IME such that the machine learning models may be updated based on selected hypotheses. The IME may utilize the updated machine learning models to process subsequent user input data to more accurately interpret the user's input at a later time. This process may repeat over time. As such, the machine learning models may become increasingly accurate.

Consider system 1 (referred to simply as "system 1"), shown below in FIG. 1 that includes an example mobile or device (referred to simply as "device 2") with an IME configured to provide multiple hypotheses using federated training, in accordance with the techniques described herein. As shown in the example of FIG. 1, system 1 includes device 2. In other examples, system 1 may include one or more devices/systems in addition to device 2. For instance, system 1 may include one or more cloud computing systems.

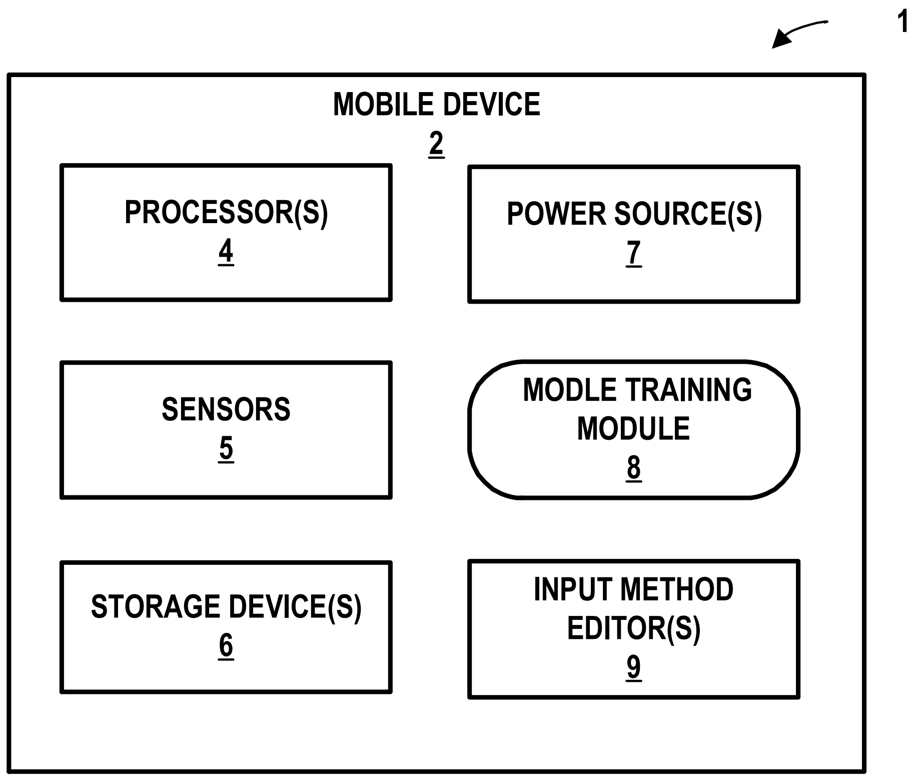

FIG. 1

While illustrated as a mobile device, device 2 may be any type of wearable or computing device. Examples of device 2 include, but are not limited to, smartwatches, visors (e.g., head-mounted computing devices), mobile computing devices (e.g., smartphones, tablets, and the like), or any other device that uses input method editors (IMEs). As shown in FIG. 1, device 2 may include processor(s) 4, sensor(s) 5, storage device(s) 6, power source(s) 7, model training module 8, and IME(s) 9.

Processors 4 may implement functionality and/or execute instructions within device 2. Examples of processors 4 include, but are not limited to, digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry.

Sensors 5 may generate data representative of various measurands. Examples of sensors 5 include, but are not limited to, touchscreen sensors, accelerometers, barometers, microphones, radar sensors, cameras, and the like. For example, sensors 5 may be a touchscreen sensor and may detect a user input, such as tapping or swiping on a virtual keyboard.

Storage devices 6 may store information for processing during the operation of device 2. As one example, storage devices 6 may store user input data received by sensors 5. As another example, storage devices 6 may store machine learning models used by device 2. As another example, storage devices 6 may store an operating system, applications, or other programs for execution at device 2. Examples of storage devices 6 include, but are not limited to, volatile memories (e.g., random access memories (RAM), dynamic random access memories (DRAM), static random access memories (SRAM), and other forms of volatile memories known in the art) and non-volatile memories (e.g., magnetic hard discs, optical discs, floppy discs, solid-state memories such as flash memories, or forms of electrically programmable memories (EPROM) or electrically erasable and programmable (EEPROM) memories, and other forms of non-volatile memories known in the art).

Power sources 7 may provide power to one or more components of device 2. For instance, power sources 7 may include a battery that provides power to processors 4, sensors 5, and storage devices 6. Power sources 7 may be rechargeable via a charger. For instance, to recharge power sources 7, device 2 may be connected to a charger via a power link. The power link may be a wired link, such as a charging cable, or may be a wireless link, such as inductive coupling.

IME(s) 9 may be executable by processors 4 to process a user's input and return multiple hypotheses in a standardized machine-readable form. For purposes of discussion, IME(s) 9 will be described being able to process and/or receive a number of different types of user inputs, such as touch and swipe typing, voice input, handwriting recognition, camera input, optical character recognition (OCR), etc. IME(s) 9 may utilize one or more machine learning models ("ML models") to perform user input recognition. The ML model may include one or more coefficients, one or more neural networks, or any other type of machine learning model.

As shown in the example of FIG. 1, device 2 may include model training module 8, which may be executable by processors 4 to train one or more machine learning models used by device 2. In other examples, model training module 8 may be located "off-device" in that such that computing resources external to device 2 can be used to perform the model training. When a user first uses device 2, IMEs 9 may implement a default model. That is, IMEs 9 may initially utilize a generic algorithm that should perform reasonably in generic cases for average user inputs.

When a user interacts with device 2 to provide user input, device 2 may save that user input data (e.g., accelerometer data from one or more accelerometers of sensors 5) in storage devices 6. Once the user provides a complete user input, IMEs 9 may use the user input data as input and return multiple hypotheses based on the user input data (e.g., provide a plurality of candidate strings to an application). Each of the multi-hypotheses may be assigned with a confidence score. For example, a user may speak the query "Despicable me" and IMEs 9 may return two hypotheses, "Speak for me" with a 0.9 confidence score and "Despicable me" with a 0.7 confidence score. IMEs 9 may select the top N hypotheses or all hypotheses with a confidence score greater than a pre-defined confidence threshold. For example, an application may set a confidence threshold to be 0.7 and IMEs 9 may determine the two hypotheses "Speak for me" with a 0.9 confidence score and "Despicable me" with a 0.7 confidence score satisfy the confidence threshold based on their confidence scores meeting the confidence threshold.

IMEs 9 may rank multiple hypotheses based on their confidence scores. In the period while this ranking is performed, IMEs 9 may display the multiple hypotheses to an application for selection. The first hypothesis can be displayed with some visual indication (e.g., highlighted with color) indicating that the interpretation is not yet considered final and is subject to change. The application may select the most suitable hypothesis, which is then considered as the correct interpretation. For example, after searching the catalog in the movie backend for the possible hypotheses in parallel, the application may find a result for “Despicable me” and choose this interpretation as the top hypothesis. IMEs 9 may now display the text that the app considered as the top hypothesis (“Despicable me”) with a hint (e.g., highlighted with a different color) to indicate that it has now been confirmed.

Model training module 8 may apply the gathered data (i.e., the stored user inputs data) to train one or more machine learning models utilized by IMEs 9. In some examples, model training module 8 may refine the models using federated learning. Federated learning allows for the hypotheses ranking to be improved based on real usage in a privacy-friendly way (i.e., without raw data leaving the device). In other examples, the ranking model can be a deep neural network which takes as input text and optionally derived features from that text and scores the input text in terms of how likely it is to be a suitable hypothesis for the application. Such training may happen locally, and gradients may be returned to a central training server which are aggregated with data collected from other user devices. This process may repeat over time. As such, the machine learning models may become more and more accurate and may be pushed to all devices which have the same application installed. In this way, model training module 8 may enable IMEs 9 to speed up the recognition of the user’s inputs, being able to generate multiple hypotheses earlier/faster.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in US Patent No. 10,332,513 B1. As another example, the techniques of this disclosure may be combined with the techniques described in US Patent No. 10,157,040 B2. As another example, the techniques of this disclosure may be combined with the techniques described in US Patent Application Publication 2020/0005081 A1.