February 2020

Indicating Remaining Battery Life in Terms of a Number of Activities That Can be Performed before Power is Depleted

Alexandru Damian

Victor Carbune

Indicating Remaining Battery Life in Terms of a Number of Activities That Can be Performed before Power is Depleted

Abstract:

This publication describes systems and techniques to indicate to a user of an electronic device remaining battery life in terms of a number of activities that can likely be performed before the battery's power is depleted. The user interface (UI) or operating system (OS) of many electronic devices, such as those of smartphones, provides an indication of remaining battery life in terms of a percentage of a fully-charged battery. For example, when a battery is halfway depleted, the electronic device indicates fifty percent (50%) remaining power. Many users are not, however, able to convert a percentage of total battery charge into a quantity of activities that can be completed before the battery's charge is exhausted. Described systems and techniques monitor recent activities performed by a user and an amount of battery usage per activity. A battery module computes how many of each of these activities can be completed based on the per-activity battery usage and a remaining battery percentage. The battery module uses the UI to indicate a number of activities that can be completed prior to the battery being depleted or reaching a critical level. In these manners, the user can tailor usage of the electronic device to ensure that important tasks are completed before the battery dies.

Keywords:

electronic device, smartphone, smartwatch, user interface (UI), operating system (OS), battery level, battery usage, battery life, percentage, state of charge (SoC), depletion, exhausted, remaining life, quantity, number, activity, action, task, event

Background:

Portable electronic devices, which include smartphones and smartwatches, operate with a battery that must be recharged when it is depleted. The user interface (UI) and operating system (OS) of many electronic devices provide an indication of the remaining battery life in terms of a percentage of a fully-charged battery. For example, when a battery is three-quarters depleted, the electronic device indicates 25 percent (25%) remaining battery life. This percentage of total battery capacity is, in many cases, too simple of a battery-level indication. This is particularly true both because of different user behaviors and because of the various power consumption patterns resulting from different devices, operating systems, and applications.

Users of electronic devices are more likely to care about how many tasks they can continue doing and for how long before the battery dies. For example, a user might want to know that there is enough battery power to send five more messages via application “A,” that it is possible to conduct searches for 20 more restaurants with a mapping application, that 60 more photos can be uploaded to cloud storage, or that the battery will likely enable one hour of additional talk time. Unfortunately, many users are not able to convert a percentage of total battery charge into a quantity of activities that can be completed before the battery charge is depleted. Therefore, it is desirable to provide users with knowledge of how many activities can be completed before the battery is exhausted.

Description:

To provide users with an estimate of how many activities can be completed before the battery of an electronic device is exhausted, a battery module is included as part of the electronic device. The battery module can be part of an operating system (OS) of the device. A battery module 110 (of Fig. 1 below), includes two components: a user interface (UI) component 112 and a battery-usage computation component 114. The UI component 112 displays battery alerts indicative of how many times an activity can be completed before the battery is exhausted. The battery-usage computation component 114 tracks battery usage and determines how many of these activities can be completed before the power is depleted. Systems and techniques are described herein first with regard to the UI component 112 and second with regard to the battery-usage computation component 114. Example approaches to implementing these components with machine-learned models (ML models) are also described below.

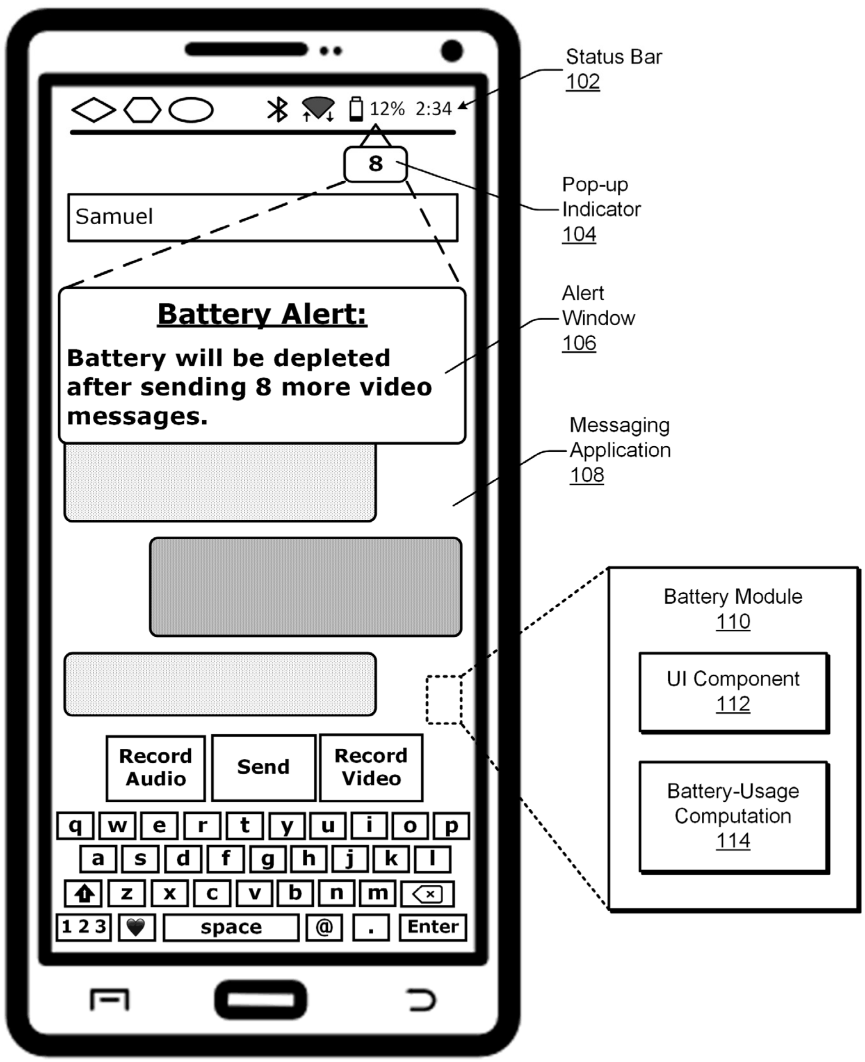

Fig. 1. Electronic device with activity-based alert about remaining battery life.

As shown in Fig. 1, an electronic device, such as a smartphone, includes a visual UI on a display screen. In this example, the device is running a messaging application 108 that is capable of recording and sending both audio and video messages in addition to sending text messages. A status bar 102 indicates various device statuses, such as running applications, wireless signal strength, and remaining battery life, in addition to the time of day. In the status bar 102, the remaining battery life is indicated as a percentage of total capacity, which is shown as twelve percent (12%) in Fig. 1.

As part of the operation of the UI component 112 of the battery module 110, the device presents a pop-up indicator 104. The pop-up indicator 104 includes a numeral indicative of a number of instances that an activity can be repeated before the battery is depleted completely or before the battery reaches some specified critical level. The user can select the pop-up indicator 104 by, for example, pressing on the icon with the numeral, which is an "8" in the illustrated example. If the user declines to select the pop-up indicator 104, the pop-up indicator 104 disappears after a few seconds.

If, on the other hand, the user selects the pop-up indicator 104 while it is visible, the UI component 112 presents an alert window 106. The alert window 106 can be displayed over other UI elements, such as those of the messaging application 108 as depicted in Fig. 1. The alert window 106 provides more-specific information about the battery alert. In this example, the alert window 106 reads, "Battery Alert: Battery will be depleted after sending 8 more video messages." However, the specific text of the alert may vary. For instance, the alert window 106 can read, "Battery will reach a critical level if 8 more video messages are sent." As another example, an alert window for a restaurant-searching scenario can read, "Search for 5 more restaurants and battery saving mode will be entered."

The battery module 110 can additionally or alternatively provide the user with remaining battery life in terms of multiple different activities, instead of merely the one activity in which the user is currently engaged. An example of this is shown in Fig. 2.

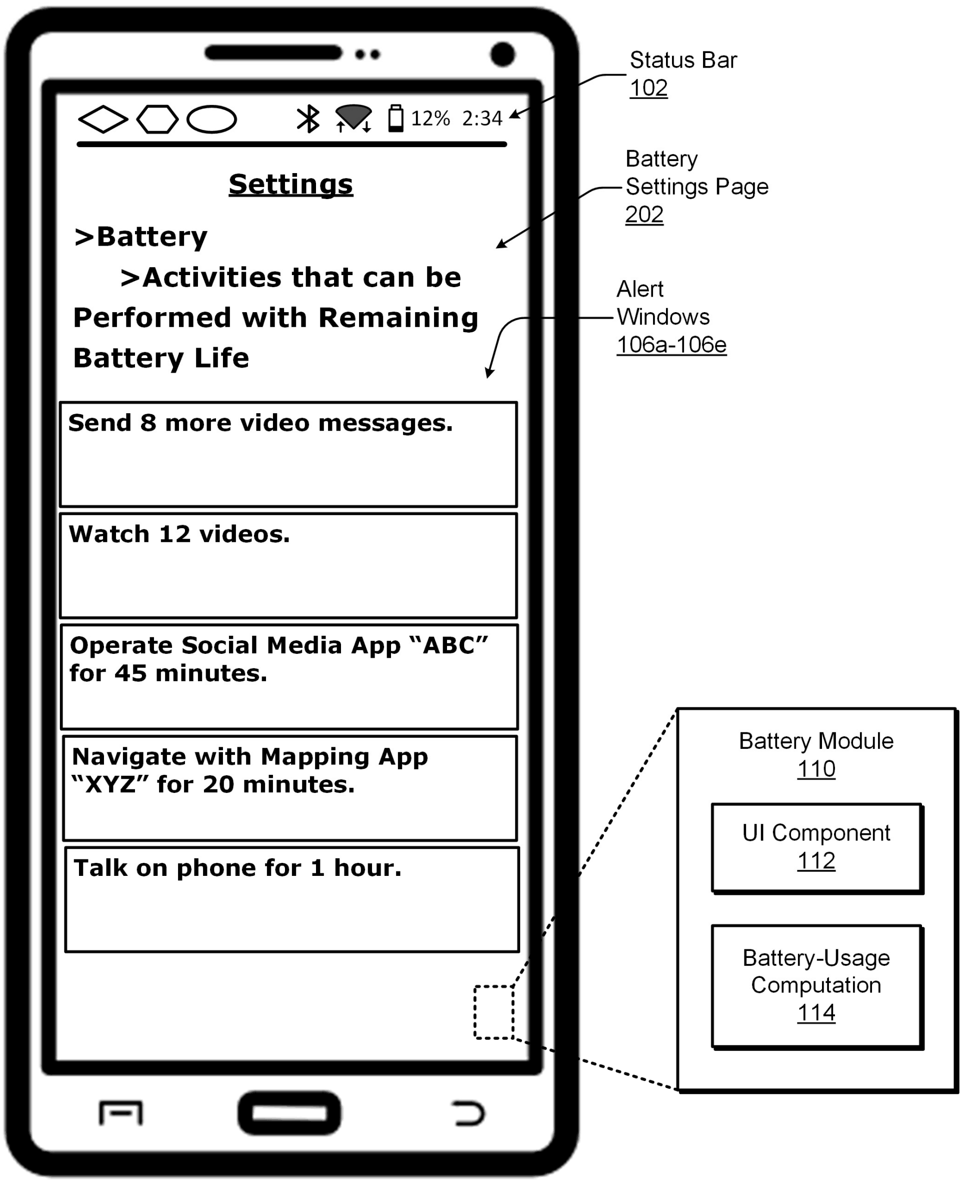

Fig. 2. User interface (UI) displaying multiple activity-based battery indications.

Fig. 2 depicts an example UI in which remaining battery life is conveyed using multiple different activity-based indications. In this example, a battery settings page 202 has been accessed, and a subpage is displayed. The UI component 112 presents indications of how many, including how many units of time (e.g., minutes or hours), of a given activity can be performed within the remaining battery life. The battery subpage includes five alert windows 106a-106e that are each presented as a tile or region of the subpage. This battery settings subpage can be accessed, for example, through a settings application of the OS or through a gesture when the pop-up indicator 104 or the alert window 106 (both of Fig. 1) is displayed. Alternatively, such a gesture may cause these alert windows 106a-106e to be displayed as a pop-up over another application.

The battery-usage computation component 114 identifies multiple activities to present as part of the alert windows 106a-106e. In some cases, the battery-usage computation component 114 selects those activities that have been used most often within some recent time frame, such as one hour or one day. The displayed activities, and the associated number of times each can be performed, include “Send 8 more video messages” and “Watch 12 videos.” Three other displayed examples are “Operate social media application ‘ABC’ for 45 minutes,” “Navigate with mapping application ‘XYZ’ for 20 minutes,” and “Talk on phone for 1 hour.”

To obtain this information for the UI component 112, the battery-usage computation component 114 tracks both the activities performed by the device and the battery usage of each of the tracked activities. Examples of this are described with reference to Fig. 3.

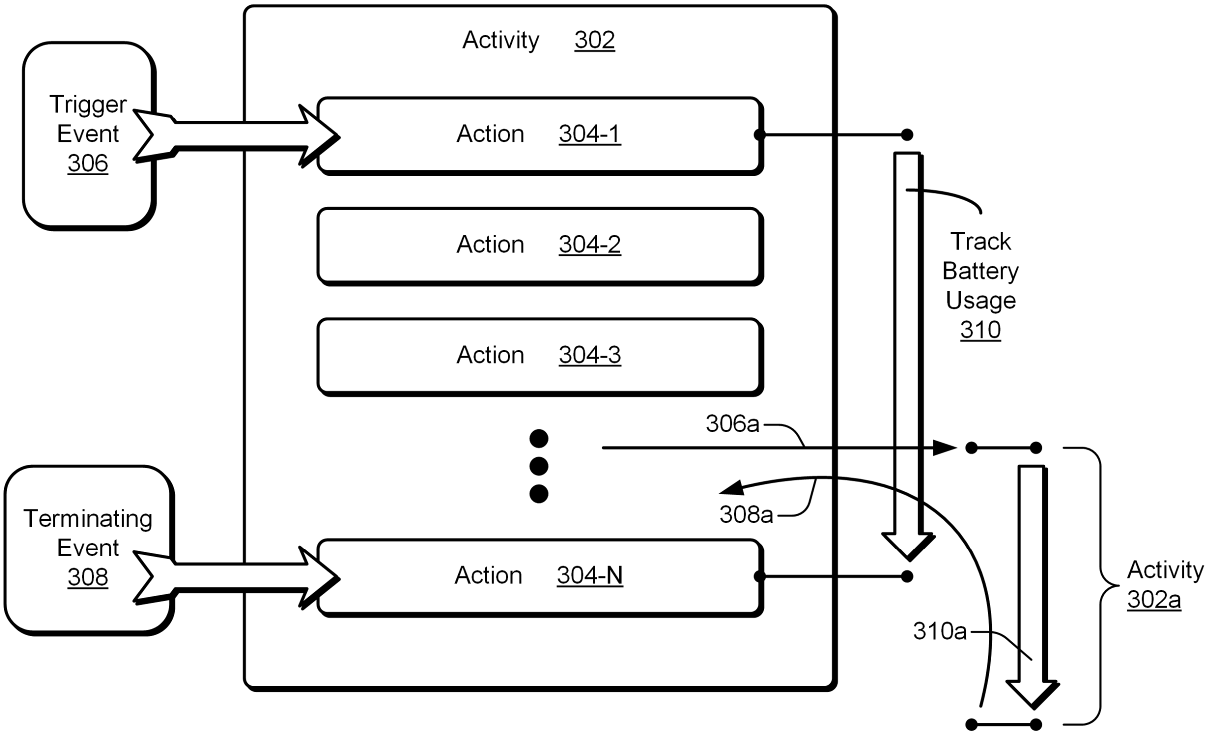

Fig. 3. Battery-usage tracking for an activity including multiple actions.

As shown in Fig. 3, an activity 302 includes multiple actions 304-1, 304-2, 304-3, …, 304-N, with "N" representing a positive integer. In some cases, each activity 302 can be manually identified, and each of the multiple actions 304-1…304-N can be manually stipulated. Examples of activities 302 include sending a message, watching a video, calling for a rideshare, listening to songs, playing a game, and so forth. For sending a message, the multiple actions 304-1…304-N can include opening the messaging application, selecting a destination, typing a message, adding an attachment, and sending the message. Initially, a default battery usage amount can be associated with each activity 302. The battery usage amount can be expressed in terms of a percentage of total capacity of the battery or in milliamps (mA).

Individual devices and users will likely have, however, different amounts of battery usage for various activities than an average or default battery usage amount. For instance, some users may attach very long videos to text messages or play games at a maximum screen brightness while other users have short videos and play with a medium screen brightness. Accordingly, in example implementations, the battery-usage computation component 114 (of Figs. 1 and 2) can update battery usage amounts by tracking actual battery usage over time on a per-user and per-device basis. Continuing the messaging example, pressing on an icon for the messaging application can be assigned as a trigger event 306. Pressing on a send button to transmit the message can be assigned as a terminating event 308. Between the two events 306 and 308, the battery-usage computation component 114 tracks battery usage as illustrated at 310.

In some environments, an activity 302 may not be triggered and terminated without starting another intervening activity. For example, while a user is entering text for a message, the user may answer a phone call, which is indicated in Fig. 3 as activity 302a. In such cases, the battery-usage computation component 114 spawns a new battery usage tracking process as indicated at 310a. This process 310a starts responsive to a trigger event 306a—such as answering the phone call—and ceases responsive to a terminating event 308a—such as the ending of the call. Once the user returns to preparing the message, the battery-usage computation component 114 continues the process 310 of tracking the battery usage amount for sending the message, which corresponds to the activity 302.

In some implementations, as described above, activities may be enumerated manually by a designer of, e.g., the OS. Additionally or alternatively, the OS may offer an application programming interface (API) for other applications to use. The API enables another application to specify a new activity 302 and optionally an initial associated battery usage amount. The other application can also provide via the API multiple actions 304-1…304-N that are included in the specified new activity. Further, the API can enable the other application to identify a trigger event 306 or a terminating event 308 (or both).

In other implementations, at least some of the functionality provided by the battery module 110 (of Figs. 1 and 2) can be automated using one or more machine-learning models. For example, a computer-readable medium (CRM) (not shown) of the electronic device can include a machine-learned model (ML model). The ML model may be a standard neural-network-based model with corresponding layers required for processing input features like fixed-side vectors, text embeddings, or variable length sequences. Additionally or alternatively, the ML model may be implemented with a support vector machine (SVM), a recurrent neural network (RNN), a convolutional neural network (CNN), a dense neural network (DNN), a regression analysis algorithm, one or more heuristics, or a combination thereof. Such an ML model can also be integrated or combined with the manual approaches described above. Example approaches to utilizing one or more ML models are described below.

To be able to display to the user "You can send 5 more messages via messaging app 'Text Best,'" the following components, which utilize at least one ML model, can be implemented. First, a model that clusters and labels activities 302 and/or actions 304 is implemented. The clustering and labeling model uses natural language to describe the types of activities most frequently performed. Second, a model that estimates battery consumption for each of these activities is implemented. The battery-consumption estimation model therefore tracks power usage and labels the tracked data appropriately. A UI model can also be implemented that decides which power usage data to show to the user based on what they need the most at any given time, e.g., what is most relevant given current tasks, how much power each task consumes, and a total remaining battery capacity.

For the clustering and labeling model, a model, which may utilize a heuristic, identifies when a user starts or stops a certain activity 302. For example, opening a messaging app, tapping on someone's name, and using the keyboard will likely indicate a "start of message sending" action. This can be implemented as a heuristic or as a classification model that is trained on labeled data (e.g., that takes into consideration the relevant features). Similarly, an "end of message" action can be recognized in the timestamp of the transmitted message as an end to the particular activity. Once these are identified, an intermediate heuristic or model is implemented that outputs whether a user is engaged in the particular activity. If the user is switching between applications and doing multiple activities at once, then the battery-usage computation 114 can, based on the model, interrupt or pause the current battery-usage tracking process and initiate another tracking process. The model can track battery usage for multiple overlapping activities as described above for the activities 302 and 302a.

In other implementations, a holistic model can be used. A holistic model is able at any given time to describe, from among a finite class of possible actions 304, what the user is doing. These actions 304 can include, for example, "start message," "open maps," "stop message," "search for restaurant," and so forth. For activities 302 for which a holistic model is impractical (e.g., too costly to create efficiently), a set of customer-facing APIs can be exposed. These APIs, as described above, can enable application developers to define a custom usage session or activity (e.g., producing a live video on a social media network). The OS can then extract relevant battery-usage signals associated with that session.

Thus, an ML model can determine one or more actions 304 for an activity 302, with the actions 304 defining a timeline of the activity 302. Another ML model tracks how much battery power is consumed for the different actions 304 based on the timeline for the overall activity 302. This can be accomplished manually, for example, by collecting data and training an initial neural-network-based regression model that outputs an estimated value of the tracked battery usage amount. The regression model can be implemented as a multi-layer neural network. The neural network takes as input the different activities 302 and/or actions 304 that have been categorized as described above and outputs an estimated battery consumption value. If a battery-usage tracking model is constantly retrained on-device, these estimates of battery-usage amounts become more accurate as the user performs more and more of the same activities 302.

After the activities 302 and actions 304 are established and the battery-tracking regression model is implemented for the battery-usage computation component 114 (of Figs. 1 and 2), the UI component 112 can display remaining battery life in terms of activities 302 that can be performed and that are of interest to the user. In one approach, the battery-impacting activities that are to be displayed are selected randomly. Alternatively, the battery module 110 (e.g., the UI component 112) can rank the activities 302 in which the user most frequently engages. As noted above, this analysis can be weighted toward activities 302 that have been performed in some recent time period. Based on this ranking, the UI component 112 can select which battery alerts to display, as shown in Fig. 2. As shown in Fig. 1, a displayed battery alert can also depend on the current context of the user. Based on the current context, the OS can infer a more-relevant battery alert message. If the user is already in a known session (like "searching for a restaurant" or "an action that uses the camera"), a single battery alert can be tailored specifically to that session. A ranking of battery-consuming activities, as shown in Fig. 2, can also place an activity pertaining to a current context at or near the top of the list.

Over time, the battery module 110 can learn additional battery-usage characteristics that are impacted by a combination of application-specific battery-usage and user behavior. As result, the UI component 112 can display comparisons between different devices or applications for the user. For example, the battery module 110 can share with the user that sending a message with application “A” is more power efficient than sending one with application “B.” The battery module 110 can also enable the user to improve battery life by suggesting what activities to perform or not perform based on which activities consume less power.

References:

[1] Patent Publication: US20160073351A1. Method of providing user with battery power notification in mobile device and mobile device therefor. Priority Date: September 4, 2014.

[2] Patent Publication: US20180143257A1. Systems and methods for estimation and prediction of battery health and performance. Filing Date: November 21, 2016.