June 01, 2017

Automatic Smart Volume Control Demultiplexing

Sandro Feuz

Sebastian Millius

AUTOMATIC SMART VOLUME CONTROL DEMULTIPLEXING

ABSTRACT

A system and method are disclosed that may predict the intent of a volume up/down event in a phone device, and direct the action so that the intended setting is adjusted. The method includes two steps. In the first step, a machine learning model deployed on the user's device is trained to predict the probability that media sound is about to be played from the device. In the second step, when the volume button is pressed, the machine learning model, depending on the predicted probability that media sound is about to be played, may delegate the action to either the media volume setting or to the ringtone volume setting. Further, the user may additionally personalize the model. The advantage of the system includes smart de-multiplexing of multiple settings through one input endpoint control. The system requires no additional system UI elements.

BACKGROUND

Handheld mobile phone devices use one volume control button to control different volume settings such as ringtone volume, media volume and alarm clock volume. The volume to be adjusted is selected depending on a set of heuristics based on what is being played at the moment or it could be based on the foreground app that could be primarily a media app. However, this fails in a lot of cases. For instance, when a user is about to play a video and would like to adjust the volume of the video by pressing the volume control button, the system would identify it as the default volume control i.e. ringtone because there is no media being played at that moment.

DESCRIPTION

A system and method are disclosed that may predict whether the volume up/down button

in a phone device will be clicked, and direct the action so that the intended setting is adjusted. The method includes two steps. In the first step, a machine learning model is trained to predict the probability that media sound is about to be played from the device. This could be done either with or without an explicit user action. In the second step, when the volume button is pressed, the system, depending on the predicted probability by the machine learning model that media sound is about to be played, may delegate the action either to the media volume setting or to the ringtone volume setting.

In the first step, training data are gathered and the gathered data are used to train a machine learning model as shown in FIG. 1A. Training data may include recordings of raw phone usage data in everyday situations, from multiple users. The gathered training data consists of a stream of phone events, such as apps being started, stopped, switched, pixels of the visible screen in regular intervals, explicit user interactions like pressing buttons, using the touch screen, audio being played etc., together with a corresponding timestamp of every event. The data are then replayed to recreate the observable state of the device at a specific point of time *t* to obtain the current phone state. For every such point of time *t*, events during the next couple of seconds (say next 30 seconds) are checked to find if media sound was played in the next 30 seconds or not and a corresponding Boolean value is generated. This may form one training sample of a user that may include the current phone state at time, *t* with a corresponding Boolean value. This step is repeated at time $\square = \square_1 \ldots \ldots . \square_\square$ to obtain n training samples for a user. The entire sequence is repeated for the multiple users, *m* as shown in FIG. 1A. This may provide a very large set of training data. The machine learning task is then formulated as "given the current phone state, predict whether sound will be played in the next 30 seconds". The system further includes sending extracted features from the current phone state as inputs to a deep neural network. The

extracted features may include ID's of the last couple of active apps, time of last user interaction, type of last user interaction, etc., as well as the raw data from the phone state such as pixels of the visible screens. The deep neural network may return as output a Boolean value of either TRUE or FALSE with a certainty probability score. The system may also use convolutional layers to extract higher level features from the raw data. Alternatively, the system may also use a recurrent neural network to input phone events as they appear, rather than recreating the total phone state at a given point of time.

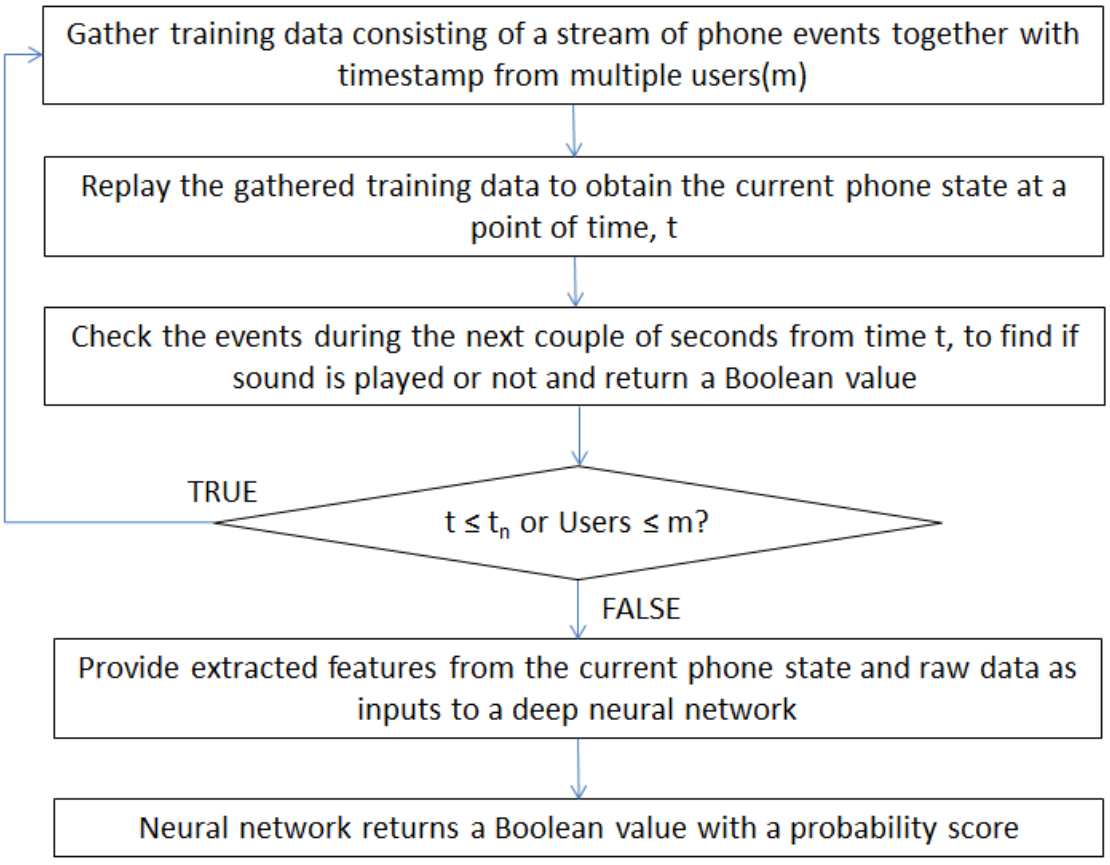

FIG. 1A: Method of gathering training data for a machine learning model

The second step as shown in FIG. 1B includes integrating the machine learning model from the first step with the user's device for the event 'when volume button is pressed'. Every time the user presses the volume up or down button, the system collects the current phone state

and feeds it into the machine learning model. The machine learning model predicts a probability score for sound to be played within the next 30 seconds. If the predicted probability score is above a certain threshold value then the system may delegate the volume up/down action to the media volume setting. If the predicted probability score is below the threshold value the system would direct the volume up/down action to the ringtone.

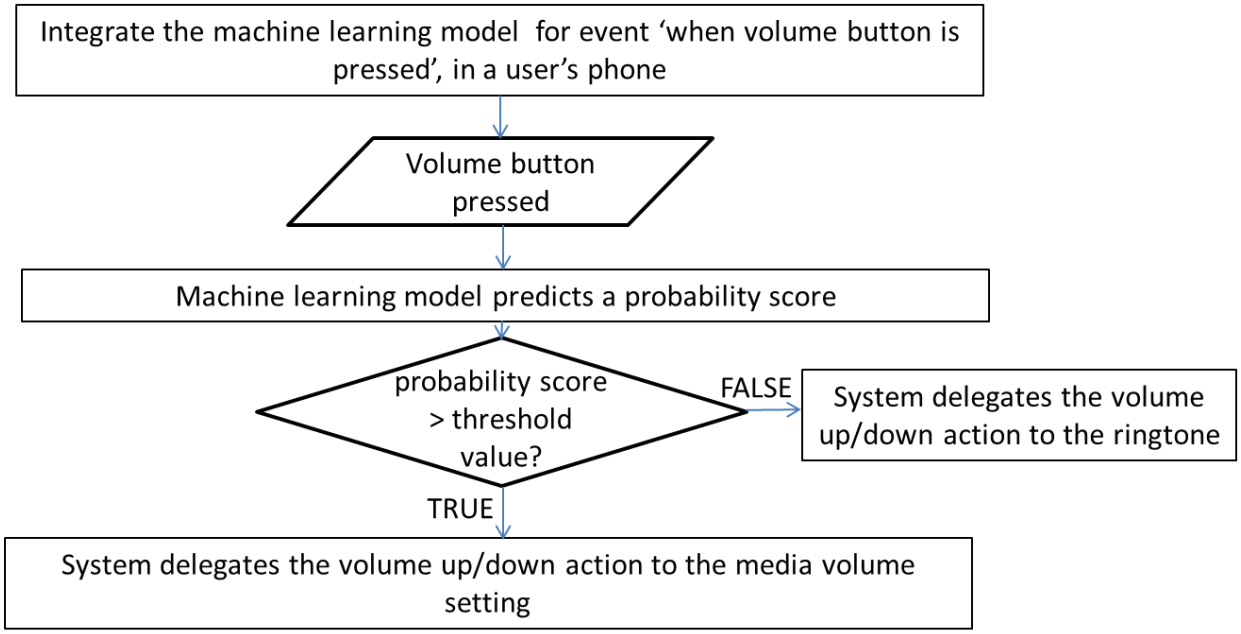

FIG. 1B: Method of selecting the action when volume button is pressed on a user device

Further, since the machine learning model is completely deployed on the user's device, the user may additionally personalize the model. Accordingly, in instances where the user overrides the system's prediction, the system may apply additional model training on the phone with those mis-predicted samples. In an alternative formulation, the machine learning model may also directly predict the volume setting to be adjusted, based on historic user usage data.

The advantage of the system includes smart de-multiplexing of multiple settings through one input endpoint control that requires no additional system UI elements or hardware extensions. The simple interface to the system allows for an intuitive interaction with the system

user.