Technical Disclosure Commons

March 02, 2018

Automatic determination of delays and appointment rescheduling

Jorim Jaggi

Yannick Stucki

Sandro Feuz

Adrian Roos

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

Automatic determination of delays and appointment rescheduling

ABSTRACT

This disclosure describes techniques to automatically predict that a user is likely to be delayed for an upcoming calendar appointment. A trained on-device machine learning model is used to predict delays based on permitted contextual information. Schedule rearrangements are automatically generated, e.g., to rearrange subsequent calendar meetings that are predicted to be affected due to the delay. Rearrangements are presented to the user, and the calendar is updated per user input and preferences. A calendar application or service that implements these techniques can reduce the burden on users to manually reschedule appointments.

KEYWORDS

- Scheduling
- Smart calendar
- Appointments
- Machine learning

BACKGROUND

It is common for users to have back-to-back meetings that are recorded in their calendars. Often, in such situations, delay in a single meeting has a ripple effect on multiple follow-up meetings which may need to be shortened, rescheduled to a different time, merged together, or canceled. Such calendar reschedule actions can be burdensome to the user.

The ability to automatically reschedule calendar appointments/ meetings with delay in an earlier calendar appointment can improve user experience of calendar applications.

DESCRIPTION

This disclosure describes machine learning models, e.g., implemented on a user device such as a smartphone, tablet, laptop, etc., that are trained for early detection of potentially delayed calendar engagements. The techniques automatically reschedule future meetings when it is detected that an earlier meeting is delayed. In this manner, cumbersome user interactions are reduced, since users need not manually edit calendar entries that are predictable consequences from delays. A calendar software application or service can implement these techniques.

A trained on-device machine learning model is utilized to enable the features. When users provide consent for use of user data to train a machine learning model, user-permitted input features that characterize the user context, e.g., user activities, location, calendar entries, calendar entries of other meeting participants that have provided consent, etc. are provided to the model. The model is also provided with training data that includes the user attendance at meetings.
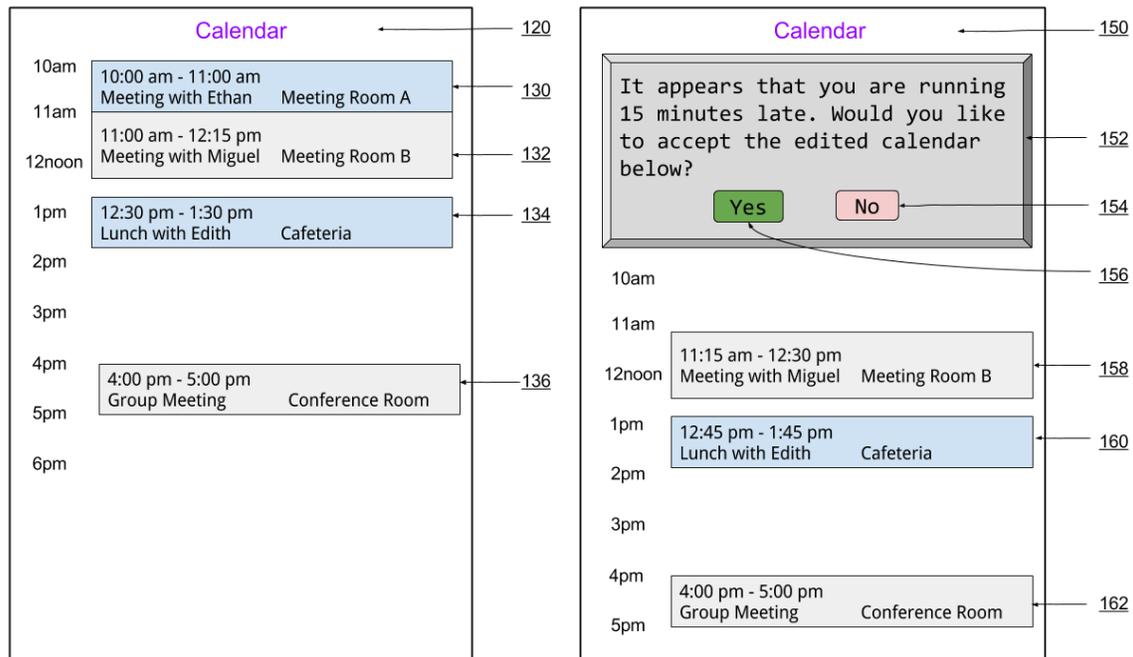

Fig. 1: Rescheduling upon predicted delays

Fig. 1 illustrates an example calendar interface on a device that implements the techniques of this disclosure. With user permission, calendar entries are automatically edited based upon a detection of a delay. In the example illustrated in Fig. 1, calendar (120) on the device includes multiple entries (130, 132, 134, and 136) for meetings. The entries include the associated time and location for each meeting.

The trained on-device machine learning model is utilized, and with user permission, obtains as input one or more of current time, sensor data, and other contextual data from the device (location information, open applications, etc.), the user's calendar entries including metadata elements such as the title of the meeting, notes in the meeting entry, participants, location, etc. Users are provided options to specify the user data that can be used as input, and can choose to restrict access to such data. Further, the automatic rescheduling features can also be turned off entirely.

With user consent, the trained model obtains such input data periodically. Based on the inputs, the model generates a prediction regarding whether the user is on schedule for upcoming calendar appointments, or if there is a likelihood of delays. The model also generates a prediction (with an associated confidence level) of whether a divergence with the events in the calendar is likely to occur. Based on the prediction, upcoming calendar appointments are analyzed to generate a score (e.g., between 0 and 1) that denotes a probability that the user is likely to be late for the upcoming appointments, e.g., by a predetermined duration threshold. The computation of the probability of the user being late for the upcoming meetings is performed, e.g., by a trained machine learning model.

In the example illustrated in Fig. 1, the user has permitted access to input data, e.g., the user's location. In this example, the time is 10:00 am, and data from the location sensor on the

phone indicates that the user is still more than 10 minutes away from Meeting Room A. This serves as an indication that the user will not be able to join the meeting in Meeting Room A on time. In the same example, if it is determined from an analysis of the location information that the meeting with Ethan (130) commenced 15 minutes late, this determination serves as an indication that the subsequent meeting at 11:00am with Miguel (132) in Meeting Room B is also likely to be delayed unless corrective action is taken. In the example illustrated in Fig. 1, it is determined from contextual location information that it will take the user 15 minutes to walk from Meeting Room B to the cafeteria. Therefore, it is determined that the lunch with Edith (134) is also likely to be delayed.

When such possible delays are detected, e.g., the score denoting the probability of delay exceeds a threshold, the user is notified. Further, automatically generated schedule rearrangements are determined and presented to the user. For example, such rearrangements are determined based on the extent of the delay and the calendar. With user approval, the rearrangement actions include cancellation of one or more meetings, postponing every upcoming meeting by a particular duration that can accommodate the delay, shortening the duration of one or more meetings, moving the next meeting to the end of a meeting block, etc.

The proposed rearrangements can be generated using various techniques, e.g., by utilizing a rules-based approach. Alternatively, or in addition, a trained machine learning model can also be utilized to determine the rearrangements. The schedule rearrangements are presented to the user, with the contextual features and the predicted delay for the first meeting. The suggested rearrangements for upcoming appointments can include suggestions to reschedule, moved, cancel, shorten the appointments, or other actions. The user can edit the suggested

actions. When users provide permission, the model can be trained using data from previous user input, e.g., acceptance of a suggested rearrangement, choice of a different action, etc.

In the example depicted in Fig. 1, a revised calendar (150) is generated. Per the revisions, the meeting with Miguel (158), and lunch with Edith (160) are each postponed by 15 minutes, while the Group Meeting (162) is left unedited at the original scheduled time. A notification is provided (152) to the user, along with options to accept (156) or reject (154) the rearranged calendar. If users provide approval for automatically accepting the rearrangements, the calendar is automatically updated. Suitable notifications and updates are also provided to other participants in the rearranged appointments.

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

This disclosure describes techniques to automatically predict that a user is likely to be delayed for an upcoming calendar appointment. A trained on-device machine learning model is

used to predict delays based on permitted contextual information. Schedule rearrangements are automatically generated, e.g., to rearrange subsequent calendar meetings that are predicted to be affected due to the delay. Rearrangements are presented to the user, and the calendar is updated per user input and preferences. A calendar application or service that implements these techniques can reduce the burden on users to manually reschedule appointments.