Reducing Distraction of Smartwatch Users with Deep Learning

Jemin Lee
Chungnam National University
Daejeon, South Korea
leejaymin@cnu.ac.kr

Jinse Kwon
Chungnam National University
Daejeon, South Korea
kwonse@cnu.ac.kr

Hyungshin Kim
Chungnam National University
Daejeon, South Korea
hyungshin@cnu.ac.kr

MobileHCI '16 Adjunct, September 06-09, 2016, Florence, Italy
ACM 978-1-4503-4413-5/16/09.
http://dx.doi.org/10.1145/2957265.2962662

Abstract
Smartwatches are overloaded with various notifications from smartphones. Users are largely distracted, while they may benefit from these relayed notification. To reduce smartwatch user's distraction, we propose an intelligent notification delivery system that relays only important notifications to the smartwatch. We claim that *important notifications* should be handled within a certain time and they are involved in launching mobile applications. To build model, we collect 6491 notifications and sensor data from three users. A mobile application has been developed to unobtrusively monitor relevant data Then, we implemented a binary classifier which identifies important notifications using deep learning and 8 features are extracted from sensor data. Our classifier shows that an important notification can be predicted with 61% - 90% and 51% - 99% of precision and recall.

Author Keywords
Notification; Smartwatch; Deep Learning.

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction

A variety of mobile devices, including smartphone, tablet, and smartwatch, are emerging and widely used. A number of users are carrying multiple mobile devices on a daily basis. Notification is a key functionality in modern wearable and mobile devices. They allow users to be aware of various events such as the arrival messages and acquaintances' activities in social networks. However, previous works have shown that notifications can distract and disrupt users who are engaged in tasks [4, 9].

Current notification delivery system pushes a notification to all connected mobile devices, including the smartwatch connected with the smartphone. For smartwatches, users can be much more distracted by the delivered notifications because they are worn almost always. Therefore, notification delivery system should relay only important notifications to the smartwatch.

In this paper, we propose a smart notification delivery system that decides whether a notification to be relayed to the smartwatch based on a deep learning model. To decide notification delivery, we first define which notifications are important ones. Previous work [10] has found that users handle the notification within a certain time and invoke related applications only if arrival time is opportune moment. From this finding, we define that important notifications are the ones that should be handled in such manners. According to this definition, our delivery system relays only the important notifications to the smartwatch.

To build the learning model, we have collected 6,491 notifications and sensor data from three users. A mobile application has been developed to unobtrusively capture the notification labels and sensor data. Then, we built a binary classifier with deep learning and 8 features were extracted from the sensor data. For predicting important notifications, our classifier attained 61% - 90% and 51% - 99% of precision and recall spanned across three users. The impact of our delivery system depends on the ratio of important notifications and the model accuracy.

User	Periods	# of Noti.	Imp. Noti. Rate
A	45	578	46.4%
B	32	1717	35.9%
C	33	4196	88.0%

Table 1: Overall information in collected data from each participant.

Related Work

The human computer interaction research groups have studied various techniques to precisely infer users' interruptibility. In desktop computer environment, they proposed Interruptibility Management (IM) system for multiple applications [4, 5]. For more accurate systems, context aware interruptibility systems were proposed [3, 14], which require a user to wear sensors for extracting context. These approaches are based on sensors attached on the human body that can trigger notifications in opportune moment by precisely recognizing context. In recent studies, researchers have exploited smartphones which are equipped with a variety of sensors for building IM system [1, 2, 8, 10–13]. As the more advanced systems, nonobstructive approaches have been presented [11, 12]. To build interruptibility models, these approaches unobtrusively monitored variation of context and system configurations without user's involvement and questionnaires. Moreover, notification's contents were considered as a context to build a better model [2, 10]. Also, OS-level IM system was designed in terms of protecting privacy and extracting deep

Group	Features
Notification's contents	Sender application name, Priority, and Title
Physical activities	Classifying activities in to six classes: InVehicle, OnBicycle, OnFoot, Running, Still, Tilting, Walking, and Unknown
Time	Time of day and Day of the week
Sensor data	Recent phone usage and Proximity

Table 2: Feature group from the collected sensor data.

context [8]. For wearable devices, Kern et al. [6] proposed a delivery mechanism which relays notifications corresponding to six contexts they defined. However, all prior works have focused on predicting opportune moment in a single device. Those works have not yet considered emerging situation in which many people carry multiple mobile devices on a daily basis. Unlike prior works, we have focused on reducing notification delivery to a smartwatch from a smartphone so that we can reduce user's distraction.

Definition of important notification
To decide whether notifications to be relayed to the smartwatch, we first define which notifications are *important*. If a notification is important, a user handles it within a certain time and lunches a mobile application that triggers the notification. Mehrotra et al. [10] defined the certain time as 10 minutes because this response time accounts for 60% of notifications. Based on this criteria, the proposed delivery system relays notifications. Other notifications are merely received on the smartphone to prevent further distraction by smartwatch.

Data Collection
In this section, we briefly describe how data is unobtrusively collected and which types of sensor data are targeted. Our research is focused on users who can use the phone and the watch simultaneously. However, the smartwatch is recently emerged. For that reason, most of general people do not have the smartwatch. In order to collect data, we limitedly distributed LG-Urbane W150 to three participants who are willing to join this project even without any monetary incentive. Table 1 lists participants and collection periods. These participants consist of 2 male and 1 female with the age span between 25 and 35 years. The data was gathered between January and March 2016 during 36 days in average.

To collect notifications, we implemented an Android application that runs on a smartphone as a background service to unobtrusively monitor notification labels (importance) and the contexts when a notification is received. For deciding important notifications, our application was built on a few APIs, `Notification Listener Service`[1] and `UsageStatsManager`[2] which are supported in API level 21 (Android 5.0). With `Notification Listener`, we can identify arrival and removal time of notifications. With `UsageStatsManager`, we can see states of mobile application usage. By combining the two APIs, our monitoring application can automatically mark important notifications. Meanwhile, our application exploits third-party library for computational social science [7], `SensorManager` and `SensorDataManager` to obtain the contexts and store a large amount of data. In addition, it uses Android OS API,

[1] https://developer.android.com/reference/android/service/notification/NotificationListenerService.html

[2] https://developer.android.com/reference/android/app/usage/UsageStatsManager.html

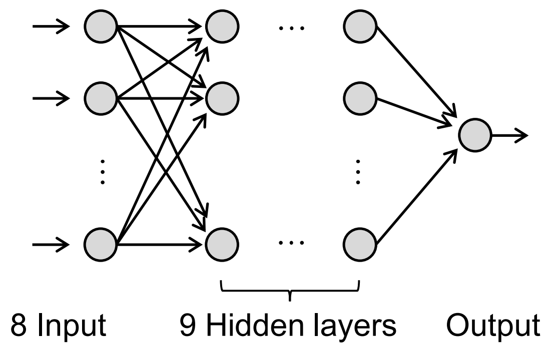

Figure 1: Fully connected 11-layer feed-forward neural network structure.

Activity Recognition[3] for monitoring user's activity. Table 2 shows collected 8 contexts by our monitoring application.

Across around 36 days, we have collected 6,491 notifications from three users. According to automatic labeling, the important notification rate of each user ranges from 35.9% to 88.0%, with the average being 56.8%. From these data, we can see that user B is distracted by most of the notifications because user B has the lowest important notification rate.

User	Precision	Recall
A	0.71	0.65
B	0.61	0.51
C	0.90	0.99

Table 3: Important notification prediction results by binary classifiers.

Building Classifier and Evaluation

In this section, we build a binary classifier to predict important notifications using deep learning. Then, we evaluate this classifier. Our model is based on our hypothesis: important notifications rely on the notification's contents and the user's context. We build a deep learning model to infer whether a notification is relayed to the smartwatch on label and 8 features.

As shown in Figure 1, designed model is fully connected 11-layer feed-forward neural network, consisting of 9 hidden layers. To train this neural network, we selected rectified linear unit (ReLU) as the activation function to avoid vanishing gradient problem. Also, 8 input features were normalized between 0 and 1. To convert continuous data into notification type like 0 and 1, the final layer of our neural network is interpreted as a logistic regression. The learning rate is 0.001 and epochs are more than 50,000 times.

For pre-processing and modeling, we used `caret` package[4] in R and Google's TensorFlow[5]. As shown in Figure 2, all users' models are converged after 50,000 Epochs.

We evaluated the neural network model by testing set. To do that, we separated training and testing sets from collected data. The ratio of training and testing data is 7:3. Table 3 shows the precision and recall of all the models corresponding to each user. Over all users, the precision ranges from 61% to 90%. The recall ranges from 51% to 99%. The poor recall leads to missing important notification by preventing delivery to smawrtwach. However, we claim that missing important notification on smartwatch is not critical because smartwatch is secondary device. Basically, users can check all notifications on smartphone even if a notification is not relayed to smartwatch.

The user B has poor precision and recall because this user prefers using messenger on computer to mobile devices. In this case, our monitoring application does not capture notification response. Finally, the monitoring application decides that this notification is not important. The poor learn-

[3] https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi

[4] http://caret.r-forge.r-project.org/

[5] https://www.tensorflow.org/

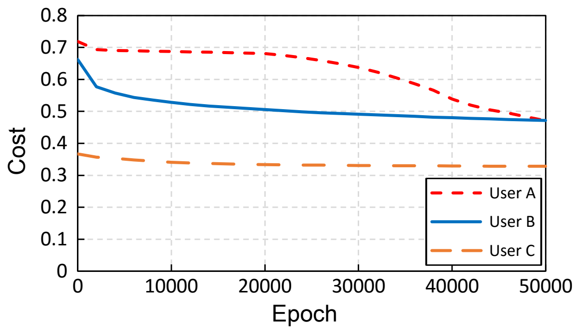

Figure 2: Converging cost function of deep neural network across three users.

ing model stems from misclassified notification label. By increasing the number of features and applying `Dropout` and `Xavier` algorithms, we can improve the accuracy of the deep neural network model.

Conclusion

Even though smartwatches improve awareness of incoming notifications, it aggravates the disruptive nature of notification delivery because this device is worn on human body. To reduce smartwatch user's distraction, we proposed a notification delivery system that relays only important notifications according to a machine learning model. To build our model, we collected 6,491 notifications and sensor data from three users using a mobile application, which unobtrusively monitors all data. Then, we implemented a binary classifier which identifies important notifications using deep learning. For important notification prediction, our classifier attained 61% - 90% and 51% - 99% of precision and recall spanned across all users. This classifier can reduce distraction of smartwatch user without noticeable degradation in users' awareness.

In the future, we are planning to deploy the proposed delivery mechanism in a real world scenario with *in the wild* notifications of frequently used mobile application. Then, we will examine real users' responses to the notification service. Also, we can improve deep neural network model by adding new features from the smartwatch. In addition, we are planning to extend our mechanism to IoT and multiple devices. By doing that, notifications can be shown on opportune devices based on users' attention.

Acknowledgements

This research was supported by the National Research Foundation (NRF) of Korea funded by the Ministry of Education (2014R1A1A2059669) and the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and NRF of Korea (2014H1C1A1066721).

REFERENCES

1. Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *Proceedings of the ACM International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*. 181–190.

2. Joel E. Fischer, Nick Yee, Victoria Bellotti, Nathan Good, Steve Benford, and Chris Greenhalgh. 2015. Effects of content and time of delivery on receptivity to mobile interruptions. In *Proceedings of the ACM International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*. 103–112.

3. Joyce Ho and Stephen S. Intille. 2005. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In

Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI). 909–918.

4. Eric Horvitz, Paul Koch, and Johnson Apacible. 2004. BusyBody: creating and fielding personalized models of the cost of interruption. In *Proceedings of the ACM International Conference on Computer supported cooperative work (CSCW)*. 507–510.
5. Stephen S. Intille, John Rondoni, Charles Kukla, Isabel Ancona, and Ling Bao. 2003. A context-aware experience sampling tool. In *Proceedings of the ACM International Conference on Human Factors in Computing Systems (CHI)*. 972–973.
6. Nicky Kern and Bernt Schiele. 2015. Context-Aware Notification for Wearable Computing. In *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC)*. 223–231.
7. Neal Lathia, Kiran Rachuri, Cecilia Mascolo, and George Roussos. 2013. Open source smartphone libraries for computational social science. In *Proceedings of the ACM International Conference on Pervasive and Ubiquitous Computing adjunct publication (UbiComp)*. 911–920.
8. Kyungmin Lee, Jason Flinn, and Brian Noble. 2015. The Case for Operating System Management of User Attention. In *Proceedings of the ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*. 111–116.
9. Luis Leiva, Matthias Böhmer, Sven Gehring, and Antonio Krüger. 2012. Back to the App: The Costs of Mobile Application Interruptions. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI)*. 291–294.
10. Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing content-driven intelligent notification mechanisms for mobile applications. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 813–824.
11. Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: designing intelligent prompting mechanisms for pervasive applications. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 897–908.
12. Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't you see my message?: predicting attentiveness to mobile instant messages. In *Proceedings of the ACM International conference on Human factors in computing systems (CHI)*. 3319–3328.
13. Stephanie Rosenthal, AnindK Dey, and Manuela Veloso. 2011. Using Decision-Theoretic Experience Sampling to Build Personalized Mobile Phone Interruption Models. *IEEE Pervasive Computing* 6696 (2011), 170–187.
14. Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Andreas Krause, Neema Moraveji, Kathryn Reiger, Jeremy Shaffer, and Fei Lung Wong. 2003. Sensay: A context-aware mobile phone. In *Proceedings of the IEEE International Symposium on Wearable Computers (ISWC)*. 248–249.