Didn't You See My Message? Predicting Attentiveness to Mobile Instant Messages

Martin Pielot, Rodrigo de Oliveira*, Haewoon Kwak, Nuria Oliver
Telefonica Research, Barcelona, Spain
*The author is currently affiliated with Google Inc., USA.
pielot/haewoon/nuriao@tid.es — oliveirar@google.com

ABSTRACT
Mobile instant messaging (*e.g.*, via SMS or WhatsApp) often goes along with an expectation of high *attentiveness*, *i.e.*, that the receiver will notice and read the message within a few minutes. Hence, existing instant messaging services for mobile phones share indicators of availability, such as the last time the user has been online. However, in this paper we not only provide evidence that these cues create social pressure, but that they are also weak predictors of attentiveness. As remedy, we propose to share a machine-computed prediction of whether the user will view a message within the next few minutes or not. For two weeks, we collected behavioral data from 24 users of mobile instant messaging services. By the means of machine-learning techniques, we identified that simple features extracted from the phone, such as the user's interaction with the notification center, the screen activity, the proximity sensor, and the ringer mode, are strong predictors of how quickly the user will attend to the messages. With seven automatically selected features our model predicts whether a phone user will view a message within a few minutes with 70.6% accuracy and a precision for fast attendance of 81.2%.

Author Keywords
Prediction; Attentiveness; Messaging; Asynchronous Communication; Availability; Mobile Devices

ACM Classification Keywords
H.5.2 Information interfaces and presentation: Miscellaneous; H.4.3 Communications Applications: Other.

General Terms
Human Factors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI'14, April 26–May 1, 2014, Toronto, Canada.
Copyright © 2014 ACM ISBN/14/04...$15.00.
http://dx.doi.org/10.1145/2556288.2556973

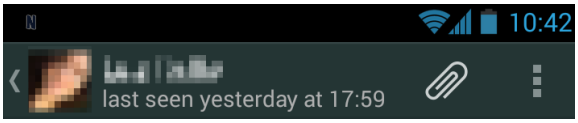

Figure 1. "Last seen" shows the time that the user had last opened WhatsApp.

INTRODUCTION
In the past few years, SMS flat rates and pervasive mobile Internet access combined with Internet-based mobile messaging applications, such as WhatsApp[1], have reduced the cost of a message to zero. These applications are used increasingly as mobile instant messengers (MIM) and users tend to expect responses to their messages within a few minutes [6, 18].

This expectation of immediacy in the communication is problematic for both the sender and the receiver of the message. For the sender his/her messages will not always be addressed within the expected time frame for a variety of reasons, including the lack of availability of the receiver. The receiver, on the other hand, increasingly feels the pressure of having to deal with dozens of notifications per day [20] and constantly checking for new messages to comply with this expectation [17].

Many instant messengers thus share cues about the user's availability. Traditionally this status is set manually, but users typically do not keep their status updated [4], and hence it becomes meaningless or creates false expectations. To tackle this issue, WhatsApp introduced what we will refer to as *last-seen time*, *i.e.* the time that the user had last opened the application, serving as an automatic approximation of availability (see Fig. 1).

This approach has two drawbacks. First, knowing when a person is online has raised privacy concerns, as it creates strong expectations: "*if you're online, it sort of means that it's in front of you and you are doing other stuff and you are ignoring me...*" [6]. Second, neither a manual availability status nor sharing the last time a person accessed a messenger are good predictors of whether the receiver will actually attend to a message soon or not. For example, the message recipient might

[1] http://www.whatsapp.com/

have just read messages before engaging in a distracting task (*e.g.*, driving, answering emails, etc.).

In this paper, we explore the value and the feasibility of answering the question: "*Will my instant message be seen within the next few minutes?*" Rather than displaying last-seen time or relying on manually-set status updates, we envision a service that automatically predicts and updates a receiver's availability to attend to instant messages (*attentiveness*).

In this paper, we define user *attentiveness* as the degree to which (s)he is paying attention to incoming instant messages. On the Android platform, attending to a message can be done in two ways: first, by opening the application that received the message, and second, by opening the notification drawer, which typically shows sender and significant portions – typically all content – of the message. At this time, the receiver gets a first idea about topic, sender, and the message's urgency. Potential social pressure, *e.g.*, the need to respond fast, begins to manifest. Further, if the message is unimportant or not urgent, the message can be discarded here.

In a survey with 84 participants, we learned that people dislike sharing WhatsApp's last-seen time, because it creates social pressure, but that they see great value in sharing at tentativeness. Thus, we developed a monitoring tool and recorded actual attentiveness and contextual factors from 24 volunteers over a period of 2 weeks. We found that seven easily-computed features can predict attentiveness with an accuracy of 70.6% and achieve a precision of 81.2% when predicting a fast reaction. In contrast, the prediction power of WhatsApp's *last-seen* status turned out to be close to random.

RELATED WORK

Instant messaging, availability for communication, and interruptibility have been extensively studied in the literature, in particular in the context of information workers and desktop computers. One of the main challenges with asynchronous communication is that when starting a conversation, time and topic are convenient for the initiator, but not necessarily for the recipient [16, 18]. In the work context, emails and instant messages have been found to interrupt workers from current tasks and make it difficult to resume it after the interruptions [1, 7].

People often **expect fast responses** when initiating instant messaging conversations: "*If I started a conversation and it's something urgent, then I expect them to respond immediately*" [6]. There is also a very fine sense towards how fast others respond: "*People do adjust their responsiveness [...] Because I try to be very responsive to people, and I expect that same responsiveness. So if they do not match up, then I am going to change my responsiveness level*" [23]. Therefore, not being able to respond fast to a message may violate the expectations of the sender and lead to tensions in the relationship. This pressure may explain why people frequently check their phones for new messages and updates [17, 21].

Sharing availability, in particular *unavailability* [18], is a way to potentially lower this pressure, as senders can adjust their expectations if they knew that the other person is busy. One of the solutions that can be found in many instant messengers is manually setting and sharing one's availability. However, the reality is that many users do not update this status reliably [4, 22].

In addition, even if people set their status to *unavailable*, it does not mean that they will not respond to inquiries. According to a recent study by Teevan *et al.* [22], people seem to even be more receptive to communication attempts received while their status shows that they are unavailable. The rationale for this unintuitive behavior is the receiver's belief that if they receive messages while being "unavailable", then the messages must be important or urgent if people contact them in spite.

Hence, relying on manually setting one's availability does not seem to be a reliable way to help users manage expectations about how fast they attend to messages.

Sharing the users' online activity is being used as an alternative. WhatsApp, for example, displays when users have last accessed the application. The idea is that this *last-seen time* would allow the sender to make an estimate of how long it will take each of their contacts to attend to their message. If a receiver has been active in the last few minutes, it may be plausible that s/he is next to the phone and hence will notice an incoming message quickly. If the person hasn't used the application for hours, s/he might be away from the phone and therefore not read a message any time soon. The study presented in this paper, however, will provide evidence that *last-seen time* is a weak predictor for attentiveness.

Further, sharing this kind of information turns out to raise privacy concerns in WhatsApp users [6]: "*people read too much into when you're online and [...] why you didn't reply and they try to guess why, and sometimes this is annoying [...] It seems like an invasion of privacy or something on the other person.*"

An alternative –not necessarily privacy-preserving– approach proposed in the literature is to provide the sender of a message with a cue of the receiver's activity. For example, Harper and Taylor [13] describe an approach which allows callers to "glance" at the recipient through their mobile phone's camera before a call, in order to see whether the person is available, which of course represents quite an intrusion into the recipient's privacy.

Predicting the reaction of the sender by the means of machine-learning techniques has been studied in the context of desktop environments.

Dabbish *et al.* [8] conducted a survey with 124 respondents from Carnegie Mellon University to investigate the features which make it more likely to reply to an email. The results show that the importance of an email is a weak predictor. Instead, people were most likely to respond to information requests and social emails.

In a diary study with 13 participants, De Guzman *et al.* [9] found that callers often desire to know more details about the potential call receiver, such as the location, or her/his physical or social availability. They suggest to share details on the receivers context prior to the call. However, as shown in [13], such approaches have privacy issues since many details about the receiver are shared.

Fogarty, Hudson *et al.* [10, 15] recorded data from four information workers to learn cues that would allow to predict when they could be interrupted in their offices. They conclude that a single microphone, the time of the day, the use of the phone, and the interaction with mouse and keyboard can estimate an office worker's interruptibility with an accuracy of 76.3%. On the basis of these findings, Begole *et al.* [4] implemented a prototype called *Lilsys*. It senses sounds, motion, use of the phone, and use of the door to predict whether an office worker is potentially available to face-to-face interruptions. Lilsys was tested with 4 colleagues and was found to help better frame their interruption, *i.e.* instead of avoiding interruptions, co-workers would start a conversation with "*I see that you are busy, but ...*"

BusyBody [14] by Horvitz *et al.* creates personalized models of interruptibility for desktop computer users, by a service running in the background and constantly monitoring computer activity, meeting status, location, time of day, and whether a conversation is detected. Tested with 4 participants, it achieved an accuracy between 70% and 87% with 470 to 2365 training cases.

Rosenthal *et al.* [19] presented a personalized method to predict when a phone should be put to silent mode. They used experience sampling to learn user preferences for different situations. The considered features include time and location, reason for the alert, and details about the alert (*e.g.*, whether it is a caller listed in the user's favorites). After two weeks of learning, thirteen out of nineteen participants reported being satisfied with the accuracy of the automatic muting.

Finally, Avrahami *et al.* [3, 2] studied the feasibility of predicting how fast a user will respond to an instant message in a desktop computer environment. In a study with 16 co-workers at Microsoft, they collected over 90,000 messages. From this data, they built models that were able to predict with an accuracy of 90.1%, whether a message sent to initiate a new session would get a response within 30 seconds, 1, 2, 5, and 10 minutes. The features included in their model were events from the instant messaging client, such as whether the message was sent or received, and events from the desktop environment, such as keyboard/mouse activities or window events. Features that were strong predictors of responsiveness included the amount of user interaction with the system, the time since the last outgoing message, and the duration of the current online-status.

Most previous work has focused on personal computers in the work environment. However, the use of these systems is typically associated with a stable context of use (e.g. work) and provides natural ways to opt-out (e.g. by not starting the messenger or by walking away from the PC). Conversely, we carry our mobile phones with us most of the day. Hence, Mobile IMs are used in very diverse contexts and are typically 'always on', keeping users engaged by means of push notifications. Hence, MIM users have greater expectations towards responsiveness, even when users are clearly unavailable, e.g. driving, in the movies, before going to bed. Thus, corpus of related work cannot necessarily be applied directly to instant messaging on mobile phones. What is missing is an extension of previous work, in particularly that by Avrahami *et al.*, [3, 2] from desktop instant messengers to mobile instant messengers (MIMs).

In view of this gap in the literature, the **main contributions** of this paper are:

1. a user survey with 84 participants to understand the perceived value and concerns of sharing predictions of availability and attentiveness,
2. a privacy-preserving machine-learning algorithm to automatically classify the user's level of attentiveness, *i.e.* whether the user will view an incoming message notification within the next few minutes or not, and
3. an extensive discussion of the importance of the tested features and the implications for the design of automatic means to share attentiveness in MIMs.

SURVEY

In order to acquire insights into people's perception about sharing availability/attentiveness in MIMs, we carried out an online survey. In particular, we were interested in feedback about three main aspects:

- How much value do people see in sharing their availability and what are their concerns?
- Could sharing a prediction of their expected attentiveness(*i.e.* "likely to read in a few minutes") be a well-accepted alternative to the state-of-the-art availability sharing, such as *last seen* used in WhatsApp?
- What should designers have in mind when incorporating such a prediction into mobile messaging solutions?

To address these questions, the survey described both approaches, *last-seen time* and a prediction of attentiveness, and asked about the subjective value and concerns regarding both of them. The survey was created with Survey Monkey and advertised via mailing lists and social networks: 102 people responded, of which 84 (19 female, 65 male) completed the whole survey, and were considered in the analysis.

Results - Quantitative

Knowing friends' availability/attentiveness

The respondents agreed to see value in knowing when a friend was last seen online ($Mdn = 4$, where 1 ='*strongly diagree*' and 5 ='*strongly agree*') as well as a prediction

of the friends' expected attentiveness ($Mdn = 4$). We did not find a statistically significant difference between the ratings ($W = 304.5, p = 0.65$).

Table 1 summarizes our participants' preferences: both types of information are considered valuable.

Preference	#	%
Prediction of attentiveness	16	19 %
Last-seen time	18	21 %
Both	35	42 %
None	15	18 %

Table 1. What would you prefer to know?

Sharing own availability/attentiveness

Regarding concerns about sharing their own availability with friends, the participants tended to be comfortable with sharing when they were last seen online ($Mdn = 3.5$) and a prediction of their expected attentiveness ($Mdn = 4$). We did not find a statistically significant difference between the ratings ($W = 466, p = 0.86$).

Table 2 shows that many respondents are comfortable with sharing information about their availability/attentiveness.

Preference	#	%
Prediction of attentiveness	23	27 %
Last-seen time	18	21 %
Both	17	20 %
None	26	31 %

Table 2. What would you prefer to share?

Results - Pros and Cons

In order to obtain strong opinions, we took a closer look at the feedback given by strong supporters of their approach, *i.e.* the respondents who had provided a clear preference for only one of the methods. In the following, we explain each of the points and illustrate it with a comment from a respondent.

Cons Sharing a Prediction of Attentiveness

Respondents preferring to share the *last-seen time* ($n = 18$) were concerned about sharing their attentiveness prediction because of

(1) being afraid to create **false expectations** ($n = 6$): "*It could be confusing and make someone not very tech-savvy misunderstand the 'probability' for a 'certainty'*",

(2) **not believing** that the method works ($n = 4$): "*I wouldn't trust 'magic figures' unless I know how they are calculated*",

(3) thinking that it's **not useful** ($n = 4$): "*won't really need that detail at all, either do or do not read but 'likely to read'?*",

or (4) having **privacy concerns** ($n = 2$): "*Depending on the source used for calculating the probability, it might reveal personal information about my future actions*".

Cons Sharing Last-Seen Time

Respondents preferring to share a prediction of attentiveness ($n = 23$) were concerned about *last-seen time* because of

(1) feeling **observed and patronized** ($n = 9$) : "*Easy prey for stalkers if you cant differ between friend and 'friend'. Too much information for certain people*",

(2) creating **social pressure** ($n = 8$): "*I might not always want to answer immediately to all messages. If this information is available [...] that puts pressure in answering or has the risk that other people thinking you're ignoring their messages*",

and (3) being the **wrong metric** ($n = 3$): "*You don't know if they have really read the unread messages*".

Pros Sharing Last-Seen Time

Respondents indicating preference to share the *last-seen time* ($n = 18$) preferred it, because of

(1) being **valuable information as message sender** ($n = 6$): "*it gives me a timeframe and allows me to estimate when my message will be read*",

(2) **communicating** to potential senders **when the user is online** ($n = 5$): e.g. "*That they can notice if I am active*",

(3) being an implicit way of **acknowledging** that a message was read ($n = 3$): "*For people I trust [...] providing a hint of their messages reaching me is relevant*",

but (4) respondents also were expressing the **wish for an option to deactivate this function** ($n = 2$): "*It would be a good idea if you can choose when to activate or deactivate it.*".

Pros Sharing a Prediction of Attentiveness

Respondents preferring to share a prediction of their attentiveness ($n = 23$) preferred this approach because of

(1) allowing to **manage expectations** ($n = 7$) "*I can spend several hours without reaching the phone. Messages are asynchronous in nature, and people should realise that. I prefer to water down expectations.*",

(2) being **curious** towards the solution ($n = 7$) "*I find it interesting to know this*",

(3) considering it **less privacy invading** ($n = 5$) "*It is less privacy-invading than knowing exact dates and times. It feels more human*",

and (4) considering it helpful to **initiate chat sessions in suitable moments** ($n = 3$) "*May help my contacts know when its better to contact me if they expect a reply*".

Implications

The quantitative responses did not reveal a clear tendency towards either of the methods. In general, our respondents had a positive attitude towards knowing their friends' attentiveness and a neutral-to-positive attitude towards sharing their own level of attentiveness.

WhatApp's model of sharing *last-seen time* was valued, because, as a sender, the respondents assumed that they could estimate when they would be receiving a response. As a receiver, it was appreciated for being an easy, implicit way of showing that one is active and reads messages. The biggest concerns were that it creates social pressure and that people feel observed and patronized. This indicates that there is a need for more privacy-preserving methods of conveying availability.

Sharing an estimate of the user's level of attentiveness was appreciated for being less privacy invading and allowing to manage expectations at the same time. The biggest concerns were that the respondents did not believe that the method would work and hence create wrong expectations. Therefore, this method would only be considered valuable –from a user-centric perspective– if it works well, and at the same time, communicates clearly that it provides an estimate and is fallible.

In summary, the survey reveals that phone users have concerns about sharing their own availability, in particular if it creates false expectations and social pressure. At the same time, over 25% of the respondents see value in sharing an estimate of how fast they are likely to view message notifications. In other words, their *attentiveness* to incoming message notifications. However, their major concerns are about whether it is feasible to accurately predict attentiveness. Hence, we investigate attentiveness prediction with state-of-the-art machine learning methods and data available on today's smartphones.

DATA COLLECTION

In order to study the feasibility of predicting attentiveness to incoming mobile messages and to explore what features could be strong predictors, we set up a study to collect ground truth data. We developed logging application and installed it on the phones of 24 Android phone users. For two weeks, we recorded both contextual data and actual attentiveness information.

Participants

We recruited participants via announcements on social networks and community forums. 24 participants (8 female, 16 male) aged 22-43 ($M = 28.7$, $SD = 5.37$) living in Europe and North America volunteered to take part in our study. Due to technical constraints (using Android accessibility APIs to intercept notifications), they had to own an Android phone with OS 4.0 or higher.

On average, participants estimated that they were receiving between 10 and 30 messages per day. *WhatsApp* was the most frequently-used mobile instant messenger, followed by *Google Hangout*, and SMS.

Asked to judge how fast they typically respond to messages, half of the participants reported to respond within a few minutes, while the other half reported to typically respond within an hour. Participants estimated that others expect them to respond within similar time frames: half within a few minutes, half within an hour.

Collected Measures

For the data collection, we developed *Message Monitor*, a background mobile service that records contextual information and message notification data on the user's phone, namely: application and time of arrival for each message, elapsed time between the time of arrival and the time of reading the message, opening and closing times of each messaging application, time when the phone's screen was turned on or off, time when the phone's screen was (un)covered (via proximity sensor), and the phone's ringer mode (silent, vibration, sound).

To learn when a message is received, Message Monitor registers as an accessibility service and intercepts notification events. If a notification belongs to a messaging app, such as SMS, WhatsApp, or Google Talk, the service logs the application and the time when the notification arrived. Note that we logged all types of notifications, and then created a whitelist-filter, containing all messaging applications that were present in the logs.

When a message arrives on Android, the phone, depending on its mode, creates a buzz and an audible notification sound. At the same time, a little icon appears on the top left part of the phone screen, the so-called *Notification Area* (see Fig. 2).

Figure 2. The Notification Area in the top left corner of the screen shows unseen notifications as little icons. The icons depict the applications, WhatsApp and SMS in this case.

No notification is generated if the messaging application is already opened. In this case, the our Message Monitor ignores the message.

In the Introduction Section we have introduced the concept of user *attentiveness* to messages, defined as the degree to which (s)he is paying attention to incoming instant messages. Ideally, we would predict *responsiveness* as done in previous work [3], but we did not do this because the act of responding is unmeasurable without instrumenting all messaging apps or the phone. Further, as confirmed by our data and previous work [8], not all messages provoke an immediate response.

To *attend* to a message the user can pull down this area and extend it into the *Notification Drawer*[2], shown in Figure 3. In the view that opens, users are provided with more details about the notifications. For short messages, the whole message can be read there. For longer messages, the user can read the subject line. Alternatively, users can *attend* to a message by directly opening the application, which has received an unread message.

When opening the Notification Drawer, we consider all unread messages as *attended* by the participant. If the user opens an application which has unread messages,

[2]See `http://developer.android.com/guide/topics/ui/notifiers/notifications.html` for detailed description

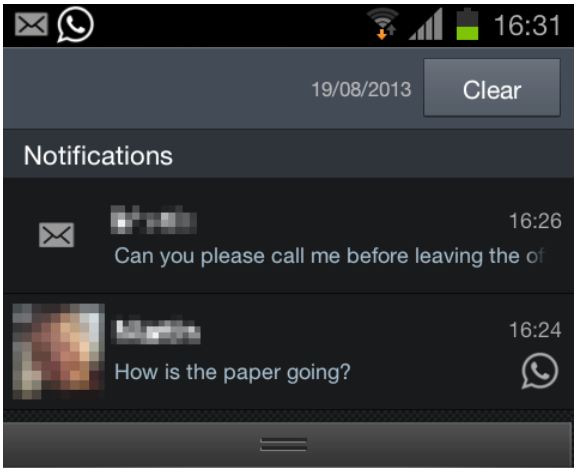

Figure 3. The Notification Drawer can be accessed by pulling down the Notification Area. This view shows details about the messages, which can include the sender and content of the message.

we consider that all messages sent to this app have been *attended*.

The status of the phone's screen and ringer were collected by registering to internal events. These events are internally fired when the respective status changes. We used the device's proximity sensor, which is located on the top of the screen next to the camera, to collect events of whether the screen was covered or uncovered.

Procedure

The study was conducted in Spring 2013. The participants installed Message Monitor and left it running for two weeks. They received 40 EUR as compensation.

Results and Feature Extraction

We collected a total of 6,423 message notifications. In average, a participant attended to a message within a median delay time of 6.15 minutes. In 74.2% of the cases, the participants first viewed a new message in the notification drawer. From within the notification drawer, they launched the messaging app in 69.1% of those cases. Thus, 22.9% of the messages were attended via the notification drawer, 77.1% via the app.

On the basis of previous work and available data, we extracted 17 potential features for the prediction of the user's attentiveness (see Table 3). The list includes features regarding, (1) the *user's activity*, such as interaction with the screen or with the phone itself, (2) *recent notification activity*, such as the number of pending notifications, (3) *the phone's state*, such as the ringer mode or whether the phone is in the pocket, and (4) the *context*, such as the hour of the day or the day of the week.

Note that we did not include any features related to the application used to exchange the message for two reasons: First, we aim at providing feedback for all messaging channels independent from the application. Second, the receiver's phone, which will estimate its user's attentiveness, does not know *a priory*, for which application to carry out this estimation, and hence would need to generate multiple statuses, one for each of the sender's messaging applications, which neither easily scales nor seems to be a usable solution.

Also note that we did not log the user's location for privacy reasons. Our focus is on capturing high-level features related to the phone's activity and its status that would preserve the user's privacy within reason. Hence, we opted against recording such privacy-sensitive data.

TOWARDS A MODEL OF USER ATTENTIVENESS

Attentiveness Classification: High or Low?

In our data collection, the median delay between receiving and attending to a message is 6.15 minutes. This is in line with the results from the recruitment survey indicating that many phone users expect responses to messages within a few minutes. Therefore, we opted to build a classifier of the user's attentiveness with two classes: *high* and *low*, with the median attending delay as pivot threshold. Hence, the problem of modeling user attentiveness turns into a class-prediction problem. Class-prediction problems have been well studied in the machine-learning community. There are many publicly available machine-learning tools to apply well established methods to specific observational records of a given problem. We have used Weka [12] for the machine-learning tasks in this work.

Classifier Selection

To solve the classification task, we tested and empirically compared the performance of a wide range of well-known classifiers, including naive Bayes, logistic regression, support vector machines (SVM), random decision trees, and random forests. We obtained the best performance with random forests, and thus used them as classifiers throughout the remaining analysis. Random forests train an ensemble of random decision trees and return the class that is the mode of the classes from the individual trees [5]. We built our random forests by using 10 decision trees.

For all tests, we randomly split the data and used 80% of the data as training data and 20% as test data. Thus, our results show how well the model can predict previously unseen data. Building a model from all features and testing it with this setup achieved an accuracy of 68.71%.

Asymmetric Error Penalization

Given that our classifier is meant to be part of a mobile intelligent user interface, it is critical to incorporate human-centric considerations when building the classifier. In the particular case of our classifier, we observe that not all misclassifications are equally *bad* from the user's perspective. In fact, as previously described in the Introduction of this paper and supported by previous work [6, 18], high expectations in terms of quick reaction times generate stress in the users of MIMs. Hence, from

No	Feature	Levels	Explanation
01	TimeSinceLastNotSec	Time (s)	Time since the last notification was received
02	LastNotifViewed	Boolean	Flag whether last notification has already been viewed
03	PendingNotCount	#	Number of unveiled notifications
04	TimeSinceLastViewed	Time (s)	Time since the user last viewed any notification
05	IsScreenOn	Boolean	Flag whether screen is on or off
06	TimeSinceLastScreenOnOff	Time (s)	Time since the screen was last turned on OR off
07	TimeSinceLastScreenOn	Time (s)	Time since the screen was last turned on
08	TimeSinceLastScreenOff	Time (s)	Time since the screen was last turned off
09	IsScreenCovered	Boolean	Flag whether screen is covered (using proximity sensor)
10	TimeSinceCoverChangedEvent	Time (s)	Time since the status of the proximity sensor last changed
11	TimeSinceLastScreenCovered	Time (s)	Time since the screen was last covered
12	TimeSinceLastScreenUnCovered	Time (s)	Time since the screen was last uncovered
13	IsInPocket	Boolean	Flag whether device is in pocket
14	RingerMode	Status	Ringer mode: unknown, silent, vibration, or sound
15	HourOfTheDay	Number	Hour of the day, e.g. '16' for 16:32
16	DayOfTheWeek	Number	Day of the week, e.g. '1' for Monday
17	IsWeekend	Boolean	Flag whether it is Saturday or Sunday

Table 3. List of features extracted from the collected sensor data.

the perspective of reducing expectations and stress, it would be better to falsely predict a slow reaction and surprise the sender with a fast response, than falsely predicting a fast reaction and keep the sender waiting, possibly leading to disappointment. In the model used above, the precision for predicting high attentiveness, *i.e.* correctly recognizing that the receiver is going to see a message within a few minutes, is 74.5%. To investigate whether we can improve this value, we explored configurations where misclassifications are not treated equally. Both for training and testing the classifier, we assigned a higher penalty cost when the *low* class is misclassified as *high* than when the *high* class is misclassified as *low*.

Such a different misclassification cost implies a trade-off between the overall classification accuracy and the accuracy for the *high* class. The higher the difference in cost between the two classes (*high* and *low*), the higher the accuracy for the fast class prediction and the lower overall accuracy. We tested the classifier by changing the relative cost of the misclassification of high from 1.0 (*i.e.* the same as that of *low*) to 2.0 (*i.e.* two times larger than that of *low*) with a regular interval of 0.1 increments. We found a significant change when the relative cost reached 1.5. Hence, the final misclassification penalty factor for the *high* attentiveness class is 1.5.

Feature Ranking for Resource Efficiency

In order to make the algorithm resource-efficient for use on mobile phones, we performed feature ranking to understand which features are the strongest predictors and to filter out redundant or irrelevant features. Using a stepwise feature-selection method, we ranked the 17 features described in Table 3. As ranking measure, we used the number of instances that are no longer classified correctly when removing a features from the full set (see Table 4, first column). Large numbers indicate high predictive power, while numbers close to 0 show that the feature has almost no predictive power. For example, when removing *TimeSinceLastScreenOn*, 8.8% less of the instances are classified correctly when compared to using all features.

Merit	Name	Acc.	Prec.
-0.088	TimeSinceLastScreenOn	0.524	0.529
-0.053	TimeSinceLastViewed	0.603	0.649
-0.042	HourOfTheDay	0.635	0.718
-0.038	TimeSinceLastNotifSec	0.622	0.701
-0.038	TimeSinceLastScreenCovered	0.672	0.75
-0.037	LastNotifViewed	0.628	0.694
-0.037	TimeSinceLastScreenOnOff	0.672	0.757
-0.036	InPocket	0.672	0.748
-0.036	RingerMode	0.689	0.763
-0.036	PendingNotifCount	0.65	0.724
-0.035	TimeSinceCoverChangedEvent	0.656	0.73
-0.034	TimeSinceLastScreenOff	0.693	0.788
-0.034	DayOfTheWeek	0.696	0.783
-0.034	ScreenOn	0.706	0.812
-0.034	Weekend	0.695	0.792
-0.033	ScreenCovered	0.684	0.759
-0.032	TimeSinceLastScreenUnCovered	0.663	0.741

Figure 4. Model performance (Overall accuracy and Precision for "High" are computed with cumulative top features)

To select an ideal subset of features for an implementation, we created a model from the feature with highest predictive power: *TimeSinceLastScreenOn*, and then subsequently added features. We added features from strongest to weakest –in terms of predictive capability– and computed accuracy and precision when classifying *high* attentiveness. If by adding the feature we improved

accuracy and precision, we kept the feature, otherwise we discarded it. Table 4 shows the features that ultimately were kept in green color, the ones that were discarded are highlighted in orange color.

The selected features comprise (1) the time since the screen was last turned on, (2) the time since the user last viewed a notification, (3) the hour of the day, (4) the time since the proximity sensor last reported that the screen was covered, (5) the ringer mode, (6) the time since the screen was last turned off, and (7) a boolean value indicating whether the screen is on or off. Using these seven features, our model achieves 70.60% overall accuracy and 81.20% *precision* for correctly classifying *high* attentiveness.

Introducing a 'very high' class

To see whether the approach could predict more classes, we tested a third *very high* attentiveness class for notifications that were viewed in less than 60 seconds – the lower quartile. This reduced overall accuracy to 61.6%, which might be too low to ensure trust in the prediction and be usable.

Comparison with Last Seen

In order to compare the achieved accuracy with cues that we find in today's mobile instant messengers, we built a *last-seen* model. It predicts the user's attentiveness on the basis of a *LastSeen* feature, which mimics the information that WhatsApp provides to its users: the time since the user last opened the messaging app. If the user is currently running the application, *LastSeen* is set to 0 in our model.

Using only WhatsApp messages and *LastSeen* as the only feature, we trained a random-forest model as described above. The resulting model achieves 58.8% overall accuracy and 53.7% of precision for the *high* attentiveness class prediction. As such, using WhatsApp's last-seen time to predict whether a user will read a message within a few minutes is almost a random guess, since by using the median response time to split the data into *high* and *low*, a random guess has an accuracy of 50%.

From the perspective of the overall accuracy & the precision for predicting the *high* class, our model with the selected features is considerably better than relying on *last seen* or random guesses. This shows that our model is a significant improvement over existing strategies.

Reflection on the Selected Features

Next, we shall discuss the selected features, which capture aspects of the user's activity, context, and habits.

Activity is approximated by *TimeSinceLastViewed*, *ScreenOn*, *TimeSinceLastScreenOn*, *TimeSinceLastScreenOff*, and *TimeSinceLastScreenCovered*. The first feature approximates whether the user has recently been viewing notifications. The screen-activity-related features are an indicator for the general use of the phone, independent from notifications or messaging services. The time since the screen was last covered is an indicator of whether there was a physical interaction around the device recently, *i.e.* has the user moved her hand in front of the screen or taken it out of the bag.

Context is approximated by the *RingerMode* and by the proximity sensor. For example, if the phone is in silent mode and in the user's pocket, new notifications will most likely go unnoticed and hence be seen with a delay.

Habits are approximated by *HourOfTheDay*. We are creatures of habit and our daily behavioral patterns are somewhat predictable [11], that is, we typically commute, work, eat, relax and sleep at around the time every day, at least during the working week. Hence, this factor captures a rough approximation of activity and attentiveness due to habits.

DISCUSSION

In sum, we have found that a data-driven model with 7 high-level features can successfully estimate a user's level of *attentiveness* to mobile messages, *i.e.* predict whether the user will attend to an incoming message within the next 6.15 minutes or not. Our model achieves 70.6% overall accuracy and 81.2% precision when predicting high attentiveness, *i.e.* the message will be seen within a few minutes. This result offers great opportunities to better manage expectations in mobile messengers.

In the following, we discuss the advantages of our proposed approach when compared to a *last-seen time* model and propose four implications for the design of mobile instant messaging applications.

Last-Seen Time versus Attentiveness Prediction

There are three dimensions where, from a human-centric perspective, there is a significant difference between currently used approaches, particularly *last-seen time*, and our proposed approach of predicting attentiveness.

Expectation Management: Previous work has argued that conveying the *last-seen time*, *i.e.* the time a messaging application had last been opened, has severe disadvantages [6], such as people reading too much into this information. Our dataset confirms that predicting whether a message notification is likely to be viewed within the next few minutes on the basis of when the user was last-seen using the application has an accuracy of only 58.8%, which indicates that expectations will be frequently violated.

Conversely, the accuracy of the proposed model represents a step forward in designing tools that help MIM users manage expectations. Unlike manual status updates, which are often forgotten to update [4], or sharing the *last-seen time*, which is a less accurate predictor, the level of accuracy that our algorithm achieves might be sufficient to make senders trust it. We plan to deploy an in-the-wild user study to shed light on the trust and value that users attribute to the proposed prediction of attentiveness. We also plan to study if this approach has a positive impact on managing expectations in MIMs.

Social Pressure: Our survey further revealed concerns of towards the *last-seen* approach, since it easily creates social pressure. For example, people cannot postpone answers in a polite manner: "*I might not always want to answer immediately to all messages. [Sharing last-seen time] puts pressure in answering or has the risk that other people thinking you are ignoring their messages.*"

We believe that it is not only positive, but crucial that the proposed model is not perfect, because it allows for plausible deniability [3] and *Butler Lies*, such as "*Sorry! I just saw your message*" [18]. Since the system will indicate that its prediction may be wrong, receivers can always blame it to an estimation error if they don't want to react in the way that the system predicted. At the same time, knowing that expected attentiveness is being shared with the sender may alleviate some of the pressure on the receiver, since the receiver won't have to explicitly explain that s/he is busy.

Privacy: Finally, *last-seen time* has raised important privacy concerns, both in our survey and previous work [6]. This is underlined by the fact that popular mobile applications have been designed to hide this *last-seen* status[3]. The features required for the computation of the proposed model, in contrast, do not require accessing personal information, and do not have to be shared with a third-party. The solution can be implemented by running a local background service on the user's phone, which monitors the phone usage and updates the prediction accordingly. If the prediction changes, it pushes a binary value (high/low) to a server, which then can be accessed by potential senders.

Despite privacy concerns, there is a certain level of desirability in users for novel solutions that help them make their receptiveness visible to others, as collected in the survey previously described: "*I can spend several hours without reaching the phone. Messages are asynchronous in nature, and people should realise that. I prefer to water down expectations.*" Predicting attentiveness may serve this desire in a privacy-preserving way.

Design Implications

Nevertheless, the survey responses suggest that a simple implementation of attentiveness prediction will not automatically be successful. User acceptance to any particular solution will depend on a number of factors.

First, it is essential to clearly communicate that the algorithm only predicts the level of user *attentiveness*, that is, if the receiver of the message will *see* the message notification within a few minutes or not. This neither means that the receiver will thoroughly read the message nor that s/he will reply. Since we found that at least 22.9% of the message did not receive an immediate reply, any implementation that fails to transparently communicate this will create false expectations.

[3]Hide-whatsapp-status `https://play.google.com/store/apps/details?id=com.hidewhatsappstatus`, last visited on Jan 5, 2014, has 500,000 - 1,000,000 downloads

Second, the service must communicate that it provides an estimate, which will be wrong in roughly $20-30\%$ of the cases. Only if this is done well, it can mitigate concerns voiced in the survey saying that senders might mistake "*'probability' for a 'certainty'*". In addition, it still needs to be empirically validated if this level of accuracy is sufficient to help users of MIMs manage expectations in their communication.

Third, the algorithm should not be perfect to allow *plausible deniability* [3]. Plausible deniability refers to a situation where the system predicts the receiver's attentiveness correctly, but the receiver decides to act the opposite way. This is particularly crucial if the system correctly predicts high attentiveness, but that the user for some reason decides to delay to respond to it. In order to take social pressure from the shoulder of the receivers, users should be able to blame false predictions to the imperfection of the system.

Finally, it might be worthwhile to investigate whether the service benefits from the user knowing the internal mechanisms of the prediction. It mitigate concerns, such as "*I wouldn't trust 'magic figures' unless I know how they are calculated*". On the other hand, it might invite to *game* the prediction. Nevertheless, there is a trend in intelligent user interfaces to allow users understand the rationale behind the intelligence, *e.g.* recommender systems explaining the reasons for their recommendations. As such, disclosing the underlying mechanism seems preferable and –thanks to the simplicity of the features– will be sufficiently easy to communicate.

CONCLUSIONS

In this paper, we have proposed a novel machine-learning approach for predicting and informing whether a person will see a phone message notification within the next few minutes (fast) or not (slow). We refer to this as the level of *attentiveness* of the user.

In a survey deployed with 84 users of MIMs, we learned that this approach is likely to be well received, given that respondents considered it to provide valuable information to both message sender and receiver in order to better manage expectations. They reported being comfortable to share a prediction of their expected attentiveness (*i.e.* "likely to read in a few minutes").

In order to verify the technical feasibility of this approach, we collected 2-week data from 24 phone users and found that not only they expect fast responses, but also that they react to message notifications with a median time of 6.15 minutes after arrival. Using the collected data and state-of-the-art machine learning algorithms, we determined that 7 easily-computed, privacy-preserving features can predict a user's attentiveness with an accuracy of 70.6% and a precision for high attentiveness (fast message viewing) of 81.2%. The selected features capture aspects of user activity, context, and user habits.

When compared to WhatsApp's *last seen* status, which turned out allow predictions not much better than random guesses, the presented approach not only offers higher accuracy, but also was commended for being less privacy invading and reducing social pressure, as informed by our survey. If designed carefully, it may strike the right balance between managing expectations and providing plausible deniability, allowing new forms of shaping communication while taking pressure off of phone users to regularly check their phones.

Longitudinal studies will be necessary to investigate if the accuracy of the system offers the right balance between trust and plausible deniability, and whether it may help to reduce impolite social behavior.

ACKNOWLEDGMENTS

We thank the people who participated in this study.

REFERENCES

1. Adamczyk, P. D., and Bailey, B. P. If not now, when?: the effects of interruption at different moments within task execution. In *Proc. CHI '04*, ACM (2004).
2. Avrahami, D., Fussell, S. R., and Hudson, S. E. IM waiting: timing and responsiveness in semi-synchronous communication. In *Proc. CSCW '08*, ACM (2008).
3. Avrahami, D., and Hudson, S. E. Responsiveness in instant messaging: predictive models supporting inter-personal communication. In *Proc. CHI '06*, ACM (2006).
4. Begole, J. B., Matsakis, N. E., and Tang, J. C. Lilsys: Sensing unavailability. In *Proc. CSCW '04*, ACM (2004).
5. Breiman, L. Random forests. *Machine Learning 45*, 1 (2001), 5–32.
6. Church, K., and de Oliveira, R. What's up with whatsapp? comparing mobile instant messaging behaviors with traditional sms. In *Proc. MobileHCI '13*, ACM (2013).
7. Czerwinski, M., Horvitz, E., and Wilhite, S. A diary study of task switching and interruptions. In *Proc. CHI '04*, ACM (2004).
8. Dabbish, L. A., Kraut, R. E., Fussell, S., and Kiesler, S. Understanding email use: Predicting action on a message. In *Proc. CHI '05*, ACM (2005).
9. De Guzman, E. S., Sharmin, M., and Bailey, B. P. Should i call now? understanding what context is considered when deciding whether to initiate remote communication via mobile devices. In *Proc GI '07*, ACM (2007).
10. Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C., and Yang, J. Predicting human interruptibility with sensors. *ACM Trans. Comput.-Hum. Interact. 12*, 1 (Mar 2005), 119–146.
11. Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L. Understanding individual human mobility patterns. *Nature 453* (2008), 779 – 782.
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter 11*, 1 (2009), 10–18.
13. Harper, R., and Taylor, S. Glancephone: an exploration of human expression. In *Proc. MobileHCI '09*, ACM (2009).
14. Horvitz, E., Koch, P., and Apacible, J. Busybody: creating and fielding personalized models of the cost of interruption. In *Proc. CSCW '04*, ACM (2004).
15. Hudson, S., Fogarty, J., Atkeson, C., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J., and Yang, J. Predicting human interruptibility with sensors: a wizard of oz feasibility study. In *Proc. CHI '03*, ACM (2003).
16. Nardi, B. A., Whittaker, S., and Bradner, E. Interaction and outeraction: instant messaging in action. In *CSCW '00*, ACM (2000).
17. Oulasvirta, A., Rattenbury, T., Ma, L., and Raita, E. Habits make smartphone use more pervasive. *Personal Ubiquitous Comput. 16*, 1 (Jan 2012), 105–114.
18. Reynolds, L., Smith, M. E., Birnholtz, J. P., and Hancock, J. T. Butler lies from both sides: actions and perceptions of unavailability management in texting. In *Proc. CSCW '13*, ACM (2013).
19. Rosenthal, S., Dey, A. K., and Veloso, M. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In *Proc. Pervasive '11*, Springer-Verlag (2011).
20. Sahami Shirazi, A., Henze, N., Pielot, M., Weber, D., and Schmidt, A. Large-scale assessment of mobile notifications. In *Proc. CHI '14*, ACM (2014).
21. Shin, C., and Dey, A. K. Automatically detecting problematic use of smartphones. In *Proc. UbiComp '13*, ACM (2013).
22. Teevan, J., and Hehmeyer, A. Understanding how the projection of availability state impacts the reception incoming communication. In *Proc. CSCW '13*, ACM (2013).
23. Tyler, J. R., and Tang, J. C. When can i expect an email response? a study of rhythms in email usage. In *Proc. ECSCW '03*, Kluwer Academic Publishers (2003).