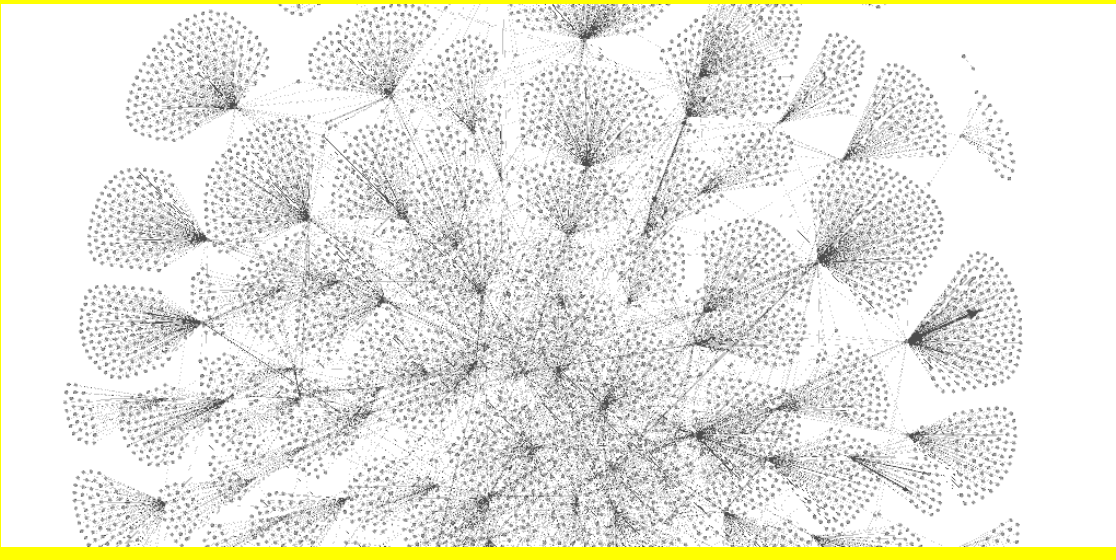
# From Signal to Social

## Steps Towards Pervasive Social Context

Jonsson, Håkan

2018

# From Signal to Social: Steps Towards Pervasive Social Context



**Håkan Jonsson**


Doctoral Thesis, 2018

Department of Computer Science
Lund University

# From Signal to Social: Steps Towards Pervasive Social Context

Håkan Jonsson
Department of Computer Science
Lund University


LUNDS UNIVERSITET
Lunds Tekniska Högskola

ISBN 978-91-7753-689-5 (print)
ISBN 978-91-7753-690-1 (pdf)
Doctoral Thesis, 2018

Department of Computer Science
Lund University
Box 118
SE-221 00 Lund
Sweden

Email: hakan.jonsson@cs.lth.se
WWW: http://cs.lth.se/hakan_jonsson

Typeset using LaTeX
Printed in Sweden by Tryckeriet i E-huset, Lund, 2018

# Abstract

The widespread adoption of smartphones with advanced sensing, computing and data transfer capabilities has made scientific studies of human social behavior possible at a previously unprecedented scale. It has also allowed context-awareness to become a natural feature in many applications using features such as activity recognition and location information.

However, one of the most important aspects of context remains largely untapped at scale, i.e. social interactions and social context. Social interaction sensing has been explored using smartphones and specialized hardware for research purposes within computational social science and ubiquitous computing, but several obstacles remain to make it usable in practice by applications at industrial scale.

In this thesis, I explore methods of physical proximity sensing and extraction of social context information from user-generated data for the purpose of context-aware applications. Furthermore, I explore the application space made possible through these methods, especially in the class of use cases that are characterized by *embodied social agency*, through field studies and a case study.

A major concern when collecting context information is the impact on user privacy. I have performed a user study in which I have surveyed the user attitudes towards the privacy implications of proximity sensing. Finally, I present results from quantitatively estimating the sensitivity of a simple type of context information, i.e. application usage, in terms of risk of user re-identification.

# Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Pierre Nugues and my assistant supervisor, Carl Magnus Olsson,for their outstanding support and inspiration. I am extremely grateful for their patience, their great enthusiasm in answering my many questions, and the many insightful and encouraging discussions I had with them. I would also like to thank Sony Mobile Communications, my employer, and my manager, Magnus Svensson, for giving me the opportunity to pursue this work as an industrial PhD student. Finally, I want to thank my wife for all her love and support.

# Preface

## List of Included Publications

This doctoral thesis summarizes my research on enabling pervasive social context. The following papers are included:

1. Tobias Ek, Camilla Kirkegaard, Håkan Jonsson, Pierre Nugues, "Named entity recognition for short text messages", in *Procedia-Social and Behavioral Sciences*, pages 178–187, 2011.

2. Tobias Arrskog, Peter Exner, Håkan Jonsson, Peter Norlander, Pierre Nugues, "Hyperlocal Event Extraction of Future Events", in *Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012)*, pages 11–20, 2012.

3. Håkan Jonsson, Pierre Nugues, "Proximates – A Social Context Engine", in *Evolving Ambient Intelligence* , pages 230–239, 2013.

4. Håkan Jonsson, Pierre Nugues, Alex Tavella, Izabela Amaral, Marina Tachibana, Vinicius Santos, "Proximity-based reminders using Bluetooth", in *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM)*, pages 151–153, 2014.

5. Håkan Jonsson, Pierre Nugues, "A comparison of two proximity networks", in *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–5, 2014.

6. Andrea Cuttone, Per Bækgaard, Vedran Sekara, Håkan Jonsson, Jakob Eg Larsen, Sune Lehmann, "SensibleSleep: A Bayesian Model for Learning Sleep Patterns from Smartphone Events", in *PLOS ONE*, volume 12(1), pages 1–20, 2017.

7. Håkan Jonsson, Pierre Nugues, "Group affiliation detection in a challenging environment", to be submitted for review to European Conference on Ambient Intelligence 2018.

8. Håkan Jonsson, Carl Magnus Olsson, “User privacy attitudes regarding proximity sensing”, submitted for review to Interdisciplinary Workshop on Privacy and Trust 2018.

9. Vedran Sekara, Enys Mones, Håkan Jonsson, “Temporal limits of privacy in human behaviour”, to be submitted for review.

## Related Patents

1. Power efficient proximity detection. US9820095B2
2. Verifying calendar information through proximate device detection. US8737950B2
3. Directional proximity detection. WO2015067982A1
4. Calendar event creation using electronic message conversations. US20120143961A1
5. Visitor detector. US8787886B2
6. Method and system for approving or disapproving connection requests. US20160337303A1
7. Adaptive media object reproduction based on social context. US9313318B2
8. Text enhancement. US8588825B2
9. User-based semantic metadata for text messages. US8849930B2

# Contribution statement

All papers included in this thesis, have been co-authored with other researchers. The author's individual contributions to papers are listed in Table 1

| Paper | Conceptualization | Data curation | Investigation | Methodology | Software | Validation | Writing |
|---|---|---|---|---|---|---|---|
| 1 | Y | Y | Y | Y | N | Y | Y |
| 2 | Y | Y | Y | Y | N | Y | Y |
| 3 | Y | Y | Y | Y | Y | Y | Y |
| 4 | Y | Y | Y | Y | Y | Y | Y |
| 5 | Y | Y | Y | Y | Y | Y | Y |
| 6 | Y | Y | N | Y | N | N | Y |
| 7 | Y | Y | Y | Y | Y | Y | Y |
| 8 | Y | Y | Y | Y | Y | Y | Y |
| 9 | Y | N | Y | Y | N | Y | Y |

Table 1: Author contributions

# Contents


1 Introduction . . . . . 1
2 Background . . . . . 2
2.1 Context and context awareness . . . . . 2
Context information processing . . . . . 4
2.2 Applications . . . . . 4
2.3 Outline of the thesis . . . . . 5
3 Related work . . . . . 5
3.1 Frameworks and technology for sensing and acquisition of context . . . . . 6
3.2 Applications of pervasive social context . . . . . 8
4 Research objectives and method . . . . . 9
4.1 Positioning . . . . . 9
4.2 Problem statement and research objectives . . . . . 10
4.3 Methodology . . . . . 11
5 Results . . . . . 12
5.1 Research Objective RO1: Develop methods for the acquisition of social context information from user-generated data . . . . . 12
Sources of context information . . . . . 12
5.2 Research Objective RO2: Assess the feasibility of physical proximity sensing for social context applications. . 16
Architectural analysis . . . . . 17
Case study: Memorit – A reminder application . . . . 18
Field study: Proximates . . . . . 23
RO2 Conclusion . . . . . 29
5.3 Research Objective RO3: Understand users attitudes on privacy with respect to proximity data. . . . . 29
5.4 Research Objective RO4: Estimate how sensitive context data is with respect to privacy, in terms of risk of user re-identification. . . . . 30
6 Conclusion and discussion . . . . . 32

6.1 Discussion . . . . . 34
6.2 Closing . . . . . 36
Bibliography . . . . . 37

# 1 Introduction

> When humans talk with humans, they can use implicit situational information, or context, to increase the conversational bandwidth. Unfortunately, this ability to convey ideas does not transfer well to humans interacting with computers.
> Cited from Abowd et al. (1999)

Our relation to machines is one of frustration (Lazar et al., 2005; Opoku-Boateng, 2015). Machines are supposed to perform and simplify our tasks, but they do not understand us like humans: We are forced to interact with them through an interface, which is neither suited for us, nor for them. To us, pressing a button has a meaning, and carries semantics. To the machine, it is just a sensor that initiates a causal chain of events resulting in some effect. Also, machines do not misunderstand us like humans either: When the button press does not do what we expect, most of us would wish to explain our intent to the machine so that it can change its action when we press again; but machines rarely listen.

With the arrival of the smartphone, and its limited size and space for interaction through controls like buttons and switches, the potential for frustration has grown. Advances have been made to find alternative ways of interaction within the field of ubiquitous computing. As a result of this, it is now commonplace for applications to adapt their content and behavior to the context of our use through indirect interactions, using our *identity*, *time*, *location*, and *activity* as inputs. For example, the Google Now application can show a user (identity) when the next bus departs (time) for his/her home when leaving (activity) from work (location).

However, applications are still oblivious to our *social context*, i.e., the people we are interacting with and our *relation* to them. Since social context is so important for most human activity, this causes frustration when we want the support of machines in social tasks, or social support in tasks executed by machines. For example, exchanging digital business cards was considered a basic future use case twenty years ago, but is still very hard to do, due to the difference in how we, humans, establish trust vs. how machines do it.

Social context is difficult to capture and make available to applications and this is the reason why it is still rarely used. The purpose of the work presented in this thesis is to take steps towards making social context available to applications, just like identity, time, location, and activity already are. The work presented takes two main approaches:

1. Extracting information about social contexts from social interactions such as text messages, or textual descriptions of social activities and places such as event websites.

2. Extracting social context from physical social interactions, i.e., being in physical proximity, in combination with online social network data and text and call communication.

The main concern with extracting this information for real-world applications is that of privacy. Therefore, I also present results on user attitudes towards social context data and on re-identification of users in high dimensional context data.

# 2 Background

The concepts of context and context-awareness are central to this work. This section introduces these concepts, especially social context, as well as the main problem areas and methods, so that the reader can understand the positioning of this thesis in the field (Sect. 4.1).

## 2.1 Context and context awareness

The concept of context has many definitions, even if we restrict ourselves to the topic of context-awareness within computer science. Dey (2001) defines it as follows:

> Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Abowd et al. (1999) also define context-awareness as:

> A system is context-aware if it uses context to provide relevant information and services to the user, where relevancy depends on the user's task.

These are very general definitions, which means that what context and context-awareness are can be highly dependent on the application and task. Thus these definitions are of little practical value when developing or discussing context-aware systems. However, in practice, there a few fundamental categories of context information that have turned out to be useful for many real-world context-aware applications. These categories are: *individuality*, *time*, *location*, *activity*, and *relations* (Fig. 1) and are part of Zimmermann et al.'s formal definition.

In this thesis, I use the operational definition of context formulated by Zimmermann et al. (2007). In addition to the fundamental categories, it provides an operational extension, which categorizes different usages and

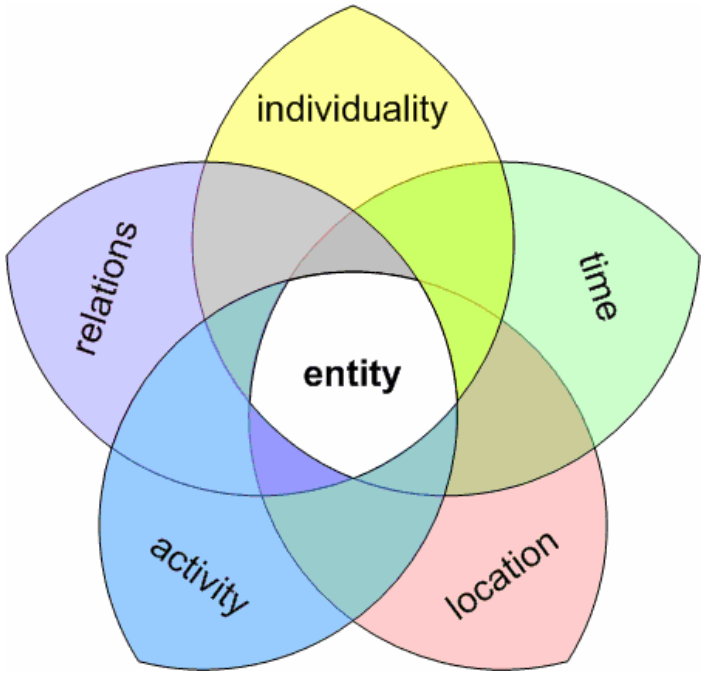



Figure 1: Five fundamental categories of context information. Reprinted by permission from Springer: LNCS, volume 4635, An Operational Definition of Context, after Zimmermann et al. (2007), © 2007

operations on context. For surveys of context models, see Bolchini et al. (2007) and Bettini et al. (2010).

Zimmermann et al. (2007) define a *social relations* sub-category to the *relations* category, which contains information about social relationships and interactions, such as Facebook friend relations, phone calls, and physical encounters. This is the main category of interest in this thesis. However, context information is by its nature collected from multiple heterogeneous sources, which are often highly dependent and usually have some uncertainty. To reduce uncertainty and to make higher-level inferences about social context, activity and location information will also be used. The phenomenological view on context considers activity and social context to be inseparable, as discussed later (Sect. 6).

When I use the term *social context* in this thesis I mean context information that includes at least the *social relations* sub-category, and may include information from the other categories. I use the term *pervasive social context* to refer to *social context* information that is sensed by means of pervasive devices, such as mobile phones and Bluetooth beacons. This

is based on the definition in the STIPI taxonomy (Schuster et al., 2013), derived from the questions who, what, where, when and why:

> Pervasive Social Context of an individual is the set of information that arises out of direct or indirect interaction with people carrying sensor-equipped pervasive devices connected to the same Social Network Service.

### Context information processing

Dealing with the complexity of context information has made context management and architectures to process context data and support management one of the main topics in context-awareness research. Another main problem is how to represent context information through modeling so that it supports reasoning and inference. Khattak et al. (2014) described the processing of context information by the following process steps: *sensing*, *acquisition*, *representation*, *fusion*, and *reasoning* (Fig. 2).


Figure 2: Context processing steps

In the *sensing* step, raw data is collected from sensors. *Acquisition* is similar to sensing but differs in the source. While sensing concerns collection of data from physical sensors, *acquisition* is the collection of information from digital sources, for example, social networks, email, or call logs, typically human-generated. *Representation* consists of transforming the acquired or sensed data in a uniform representation, typically using an ontology, an object model, or another formal model such as Zimmermann's. *Fusion* is a step, where aggregation operations are performed. This can mean aggregating data from a single source to reduce uncertainty or combining multiple sources to a higher level abstraction. Finally, *reasoning* is where inferences are made based on the data. When the inferences have been made, we can use them in applications. So what applications can we create using social context?

## 2.2 Applications

When applications have access to social context information, new use cases become possible, for example:

- Adapting content recommendations depending on whether the trip a user is currently making is a vacation or a work trip, based on whether the user is with his/her family or with colleagues;
- Adapting movie recommendations depending on the preferences and past viewing history of the people in an *ad hoc* group;
- Sending a text message to the people who attended a meeting, rather than those invited;
- Automatic tagging of people present in a photo a user takes;
- Reminders based on when a user meets someone, for example, to pay back lunch money he borrowed;
- Automatically turning off phone ringtone signal when in class or meeting;
- Automatically exchanging business cards with everyone we have business meetings with.

These are use cases where we need our technology to represent us in the sense that we are discoverable in the physical proximity of technology representing others. Technology that can represent us in this way can be said to be capable of *embodied social agency*. I define and elaborate on this concept in section 6.1.

Designers of products rarely know anything about the social context of use of applications before releasing the application. With the new capabilities of connected products with sensors, capturing the social context of use will allow for a better design of products.

### 2.3 Outline of the thesis

After this introduction, I survey previous related research in context awareness and social context modeling (Sect. 3). Section 4 defines the research problem and questions and how they relate to existing research questions in the field. Section 5 contains an overview of the included papers, their results, and how they relate to each other and the research objectives.

## 3 Related work

In this section, I survey previous related research in context awareness and pervasive social context.

## 3.1 Frameworks and technology for sensing and acquisition of context

Several context frameworks for applications have been developed. Early frameworks targeted desktop applications, for example (Dey et al., 2001) and focused on capture and triggering of application events. These were followed by frameworks intended to developed context service infrastructure through interfaces and protocols, for example Java Context Application Framework (Bardram, 2005). In more recent years the focus of frameworks such as JigSaw (Lu et al., 2010), Nobodo (Bell et al., 2011), LDCC (Kiukkonen et al., 2010) and Funf (Aharony et al., 2011) has been on sensing, acquisition, fusion and representation in mobile phones, often supported by a cloud infrastructure for storage. The reason that mobile phones is the target environment is that this allows for large scale deployments and in situ studies. AWARE (Ferreira et al., 2015) is the most comprehensive such framework, building on the experiences drawn from earlier frameworks such as Ohmage, CORTEX, Context Studio, and Funf. It provides sensing of software and hardware sensors, acquisition of user-generated data (called human-based sensing in AWARE), representation, plugins, user privacy controls and a cloud services framework. The main advantage of AWARE is its plugins and open source availability. This have attracted a quite large community of researchers. Another interesting feature of AWARE is that it incorporates support for *experience sampling* (ESM), i.e. explicit user information about experiences. This allows for contextually triggered ESM-questions rather than at random or fixed times during a study.

AWARE claim to address the needs of researchers, application developers and user alike. However, AWARE is installed as availability service, allowing it to take control over low level services. This is a security risk, and Google recently changed the accessibility services API to prevent this. Thus AWARE is no longer allowed to be distributed in Google Play.

AWARE and some other frameworks, e.g. Funf, LDCC and Nobodo, support collection of beacon and Bluetooth which can potentially be used for capturing physical proximity. CenceMe (Miluzzo et al., 2008) specifically support social networks data, but for the purpose of sharing context data to these social networks, rather than mapping between physical proximity networks and online social networks or call networks.

The frameworks mentioned above are all software-based. Hardware platform designed specifically for collecting social interaction research data with high resolution and accuracy using a broader range of sensors have been developed. SocioPatterns (Barrat et al., 2008) is a RFID-based platform for social interaction sensing. The Live Social Semantics application (Van den Broeck et al., 2010) use it in combination with online social network data to study social behaviors at academic conferences.

SocioPatterns is similar to the Sociometric Badge (Olguín et al., 2009) hardware developed at MIT by the Human Dynamics group, that combines Bluetooth, Wifi, IR and voice sensing to capture social interactions. Both of these platform have successfully been used in several studies of human social behavior in real-world situations (Atzmueller et al., 2014; Génois et al., 2015; Szomszor et al., 2011; Olguín et al., 2009). Montanari et al. (2017) developed a wrist-worn specialized hardware to collect Bluetooth proximity information for social interaction sensing. None of these are suitable for longitudinal or large scale studies, or for development of applications for production deployment.

The SDCF framework (Atzmueller and Hilgenberg, 2013), a software framework, claims support for social network data, but that is only in the form of general virtual sensor support, and it does not provide any implementations to deal with the complexities of dealing with social network APIs. The published source code does not support Bluetooth. In (Atzmueller and Hilgenberg, 2013) the SocioPatterns RFID-based badges are used as ground truth for Bluetooth proximity to detect social interaction, reaching similar results as we do in Paper 7.

A major problem for all frameworks that target phone operating systems is power consumption. Sensing consumes power, and frequent collection of sensor data requires frequently waking up the application CPU. A common approach for Funf, LDCC and Jigsaw is that they try to determine the phone sensing context in order to reduce power consumption. Determining a detailed phone sensing context is not trivial (Miluzzo et al., 2010) and consumes power in itself. Application level frameworks suffer from this problem due to lack of access to low level system APIs in Android or other phone operating systems. They also lack the the resources required to make adaptations to different phone HW platforms.

In 2016 Google introduced the Awareness API in Android (Google, 2016), solving the many of problems these frameworks try to solve. In a single and simple API, information about location, place, weather, activity nearby beacons, and headphones state is provided through polling or events. This means there is a standard format and API for basic context information on Android. More importantly though, the Awareness API manages all low level sensor fusion and hardware and platform differences, as well as minimization of power consumption.

While performing the research presented in this thesis, I worked at Sony Mobile, an Android phone vendor. This gave me the opportunity to give feedback and requirements to Google based on my research during the development of the Awareness API.

Google has also implemented an API in Android called Google Nearby API (Google, 2015). It solves the problem of detecting nearby devices using

a combination of Bluetooth Low Energy (BLE) and ultrasound. The API makes it very easy to write applications that interact with nearby devices, and also makes the user interaction simple. However, it only addresses device interaction and simplification of configuration and interaction. It does not address the problem of how our devices can represent us both in the physical and digital world.

## 3.2 Applications of pervasive social context

There have been many studies performed within ubiquitous computing and computational social science to understand human social behaviour in various settings. This was largely initiated by Nathan Eagle and Sandy Pentland when they started their work on what they call *reality mining* (Eagle et al., 2009). Using mobile phones, Eagle collected sensor data including Bluetooth proximity data, call information and survey data. From this data, social network structures and behavior patterns were extracted, resulting in a 95% prediction accuracy of friendship using sensor data. The Human Dynamics group then developed and used the Sociometric Badge in several in-situ studies of social behavior in organizations. These studies showed how analysis of such data can be used to predict workplace satisfaction, communication quality (Olguín et al., 2009) and productivity (Wu et al., 2008).

Stopczynski et al. (2013) studied social interaction behaviors in participants of a music festival, showing how social interaction data can enhance the festival experience and future planning of events.

The studies listed above all perform the analysis after the study has been done, in order to study social behavior for research purposes and potentially propose interventions for improvement. They do not target the development of context-aware applications that make use of the data collected in the situation.

Using the data collected in the Lausanne Data Collection Campaign (LDCC), Do et al. (2011) were able to identify relations between application usage and context, resulting in design suggestions for supporting synchronous communication and context-dependent offering of functionality on mobile phones.

There has been some work done on using pervasive social context for context aware applications. Social Serendipity (Eagle and Pentland, 2005), developed by Nathan Eagle, combined proximity sensing with user interest profiles to perform interest base-matchmaking, recommending users in physical proximity of each other and having matching interests to socialize.

Live Social Semantics was one of the first applications to be used to study social interactions between conference attendees, showing how physical interactions can be combined with online social network information to

facilitate social interaction.

More recently, dating apps developed along the lines of Social Serendipity have been published, for example YAC (YAC, 2017), that allows its users to detect other YAC users who match their dating profile, using Bluetooth to detect proximity. Even though Social Serendipity was the first such application on mobile phones, the Japanese company Erfolg developed specialized hardware called Lovegety (Iwatani, 1998) for the same purpose already in 1998.

# 4 Research objectives and method

In this section, the research problem and objectives are stated, and the methodology used to accomplish them is explained.

## 4.1 Positioning

Since making it possible for applications to use social context is a relatively unexplored area, the work presented here is exploratory, investigating several different options and prerequisites for enabling social context to applications, as well as consequences for the user of doing so. For this reason, the work presented is interdisciplinary. This thesis is positioned within ubiquitous computing, and more specifically within context-awareness.

There has been considerable research done in context-awareness. As illustrated in Figure 3, the work in context representation, fusion and reasoning is horizontal in the sense that it aims to be independent of domain and application. Apps are then built on top of this as proof of concept. Vertical apps are also common in ubiquitous computing, where the aim is to research a specific topic, and not to generalize across applications or domains. Sometimes applications are research artifacts in these cases, but they are usually vertical, i.e., they are not designed to study methods for social context for other applications.

In computational social science and related areas such as reality mining, human dynamics and social physics, mobile phones are used in research to investigate human social behavior. The goal is then to find fundamental insights into social and human phenomena, and not to build applications, especially not context aware applications. To measure physical social interactions in these studies, custom HW is often designed. However, these are usually not suitable for longitudinal studies since they are not designed for this and are hard to maintain in the field. Also, since they are custom built, it is hard to scale them for studies with larger populations.

Thus, there is a gap in research on social context: Feasible methods for making social context available to applications at scale, and exploration of

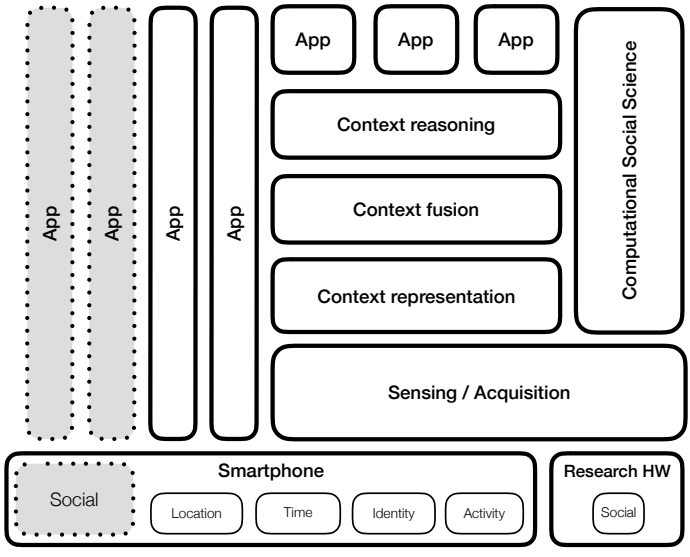



Figure 3: Research gap

what applications are made possible by use of social context (grey area in Figure 3). The purpose of this thesis is to address this gap. By feasibility we mean that the methods need to fulfill the following three requirements:

- General application requirements;
- Application requirements on social context;
- Privacy-related requirements.

## 4.2 Problem statement and research objectives

Context acqusition involves many data types with varying granularity, noise levels, and dependencies. Thus, the practical aspects of getting each of the modules to work, representation, fusion, and reasoning, is a research issue in itself, rather than just an engineering exercise. As stated by Moran and Dourish (2001):

Due to the complexity of context awareness, the practical aspects of getting context awareness to work is a research issue in itself, rather than just an engineering exercise, as stated by Moran and Dourish (2001):

> Context awareness is fine in theory. The research issue is figuring out how to get it to work in practice.

Much of the problem in context awareness consists of collecting and inferring context, as complete as possible when data comes from multiple heterogeneous sources and types with varying degrees of uncertainty, freshness, and abstraction. The *relation* context (and especially its sub-category, social context) is not as readily available as the *time*, *identity*, *location*, and *activity* categories.

The general research question addressed in this thesis is the following:

> How do we sense and infer social context, and make it available to applications in a way that allows us to balance between simplicity, accuracy, utility, and privacy while scaling to millions of smartphone users?

I address the question above by four research objectives:

RO1 Develop methods for the acquisition of social context information from user-generated data (acquired context).

RO2 Assess the feasibility of physical proximity sensing for social context applications.

RO3 Understand users' attitudes on privacy with respect to proximity data.

RO4 Estimate how sensitive context data is, with respect to privacy, in terms of risk of user re-identification.

## 4.3 Methodology

The main methodology used in this work resorts to *design science* (Hevner et al., 2004). To address research objectives RO1, RO2 and RO3, artifacts are designed and then evaluated. Six papers are included, of which four investigate separate artifacts, all relating to the goal of enabling the use of social context for applications. RO4 is addressed in an analysis using data from an available dataset and does not use design science.

Mobile phones and smartphones have proven to be an invaluable tool for social research, as shown by Raento et al. (2009). It is commonly used in computational science, social complex systems research, and ubiquitous computing for its ability to reach large number of test subjects and collects a wide range of social behavior data such as location, communication and interaction data.

Smartphones are used as the main tool in thesis too, for two reasons: The first reason is the one just mentioned above. The other reason is that

smartphones are also one of the very few target environments for applications that scale to a large number of users, which is our research objective (RO2).

In Sect. 5, we use Zimmermann's context model to show how each paper presented contributes to our understanding and use of context. The main reason for choosing the operational definition of context by Zimmermann et al. (2007) is that it was specifically created to bridge the gap between users and developers of context-aware applications. It provides a general definition, a formal definition of context information categories, and an operational extension characterizing use of context.

# 5 Results

This section presents methods and results from the papers included and corresponding studies, structured by research objective. I use Zimmermann's model to categorize the context information acquired or sensed in each study, and how each result contributes to the overall context information picture.

## 5.1 Research Objective RO1: Develop methods for the acquisition of social context information from user-generated data

### Sources of context information

We can acquire social context information from several sources. A user's calendar is an obvious example and was investigated by Khalil and Connelly (2005). A calendar entry often provides information that matches several leaves of Zimmermann's context model (Figure 1): The *time* of the meeting, the *location*, such as a room or an address, the subject to be dealt with (*activity*), the attendees, their identities and organizational belonging (*individuality* and *relations*).

Like all sources of context, a calendar has uncertainty associated with it: Some information may be assumed to be known by the attendees and thus left out, for example, the location of the office, if it is an office where all the attendees work. Also, not all users use an electronic calendar or store it in their phone. Even if they do it, it is common to have several meetings in the calendar that are never attended. Also, calendar entries of meetings are often used for reminder purposes, intended only for the owner of the calendar. If the entry is not shared, the user often already knows where the meeting will take place, and who will attend, and thus is not entered. Additional sources of context information may reduce uncertainty

in inferring social context. Thus, we want to collect context information from as many sources as possible. To summarize, a calendar alone does not provide enough information with high enough certainty to rely on it as the only source of information for context-awareness.

In the following section, I present results on acquiring context information using user-generated semantic data in the form of text, and on acquiring physical and social activity data by proxy of user phone activity.

**Named entity recognition for short messages.** Battestini et al. (2010) showed that 32% of all text messages we send are used to plan future meetings. Such planning usually includes exchanging information about when and where to meet, with whom, and why. This makes text messages a valuable source of information about future social contexts if it can be properly extracted. In natural language processing (NLP), this type of information is called *named entities* and the process for extraction is called *name entity recognition* (NER).

To address RO1, I developed a named entity recognizer artifact, which was subjected to a *dynamical analysis* to evaluate its information extraction accuracy. In terms of Zimmermann's context model, the method aims to acquire context information in the *time*, *location*, *activity*, *individuality*, and *relations* categories, thus potentially contributing significantly to reducing uncertainty through context information fusion as discussed in Sect. 2.1

Paper 1 presents the method and artifact developed for analyzing the content of texts messages to extract time, date, place, names, and phone numbers. The named entity recognizer uses machine learning. More specifically, it consists of an ensemble method, combining regular expressions and a logistic regression classifier.

The named entity recognizer was applied to a corpus of Swedish SMS text messages resulting in an extraction F-score of 86%. This accuracy was better than the only previous published results (Jiang et al., 2010) on NER for SMS messages. Compared to Jiang et al. (2010), we achieved a higher accuracy at the cost of higher memory usage, and targeting a different language. Since then, several studies have been made on named entity recognition applied to tweets. Named entity recognition on tweets are potentially similar, but according to af Segerstad (2002), these linguistic features of SMS are different from those of other text media. Comparisons to linguistic features of other messaging applications and types are hard to make since these rarely provide any means of extracting the text from the app.

In addition to the named entity recognizer, an application artifact was prototyped in a *case study*. The application allowed users to see what information had been extracted from his or her text messages, that could be

used to create calendar entries. The application was tested on a small user group of 6 people, who were interviewed about their experiences. The main conclusion from this qualitative investigation was that the users perceived that the information was mostly correctly extracted. However, it failed in reconciling information across multiple messages or conversations extended over time. The extraction of the context information described fulfills research objective RO1 since it is only intended to cover the acquisition of context information, not its fusion.

This work resulted in two granted patents, both how to improve messaging UX using NER to extract contextual information (Jonsson, 2013) (Jonsson, 2014a).

**Hyperlocal event extraction of future events.** In developing the named entity recognizer above, I found that many of the locations extracted had folksonomic names, and were often ambiguous. Names such as "the central" probably refer to a central station, but in which city? Without further information about location, for example city-level position of the sender or receiver, these are hard to disambiguate to a specific location, and thus hard to fuse with other context information.

*Activity* and *location* are crucial components of social context and thus important context information categories, since knowing the location can help us understand what activity is taking place there, and what type of social interactions. Event databases (e.g., Eventful, Zvent) and location databases (e.g., Foursquare) can be used to determine the semantics of a place and the current and future activities taking place there.

However, these databases have small coverage and mostly include major commercial events in densely populated areas. Most smaller events and events outside major city areas are announced in local media, on local websites, on private Facebook groups, mailing lists, etc. in unstructured or semi-structured text format. We call these events *hyper-local*. In Paper 2, I explore the possibilities of extracting future hyper-local event information from a wide range of web sources, to determine if the document structure and content can be exploited for human resource-efficient event scraping.

The paper describes two experimental knowledge-driven, pattern-based programs that scrape events from web pages using both their content and structure. The information extracted contained the event title, date, time, and geographic location, fitting in the *activity*, *time*, and *location* context information categories of Zimmermann's model. The simpler method of the two achieved an F-score of 72% with an average setup time of 34 minutes per site, while the more generic one achieved an F-score of 60%, but only requiring an average setup time of 12 minutes per site.

The hyper-local event extractor exposed a SPARQL endpoint that was

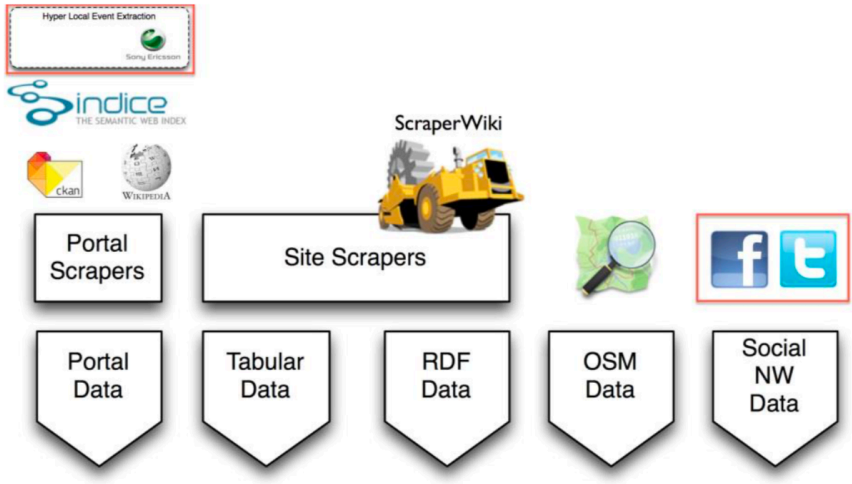



Figure 4: WED-pipe acquisition layer

Figure 5: A user enjoying the augmented Bastille view.

integrated into the Sindice semantic web indexer and used in the Venturi project social data mining component called WED-pipe (Mostarda, 2014) as part of the final demonstrator system (Giulio and Chippendale, 2014) (Figure 4).

Venturi was a European Union funded Project in the 7th Framework Program, focused on researching context-awareness for Augmented Reality. The event content was extracted from local sites in Grenoble, and shown in the demonstrator application. From the Bastille scenic viewpoint in Grenoble, users of the system could see an augmented view of Grenoble containing localized event information (Figure 5).

The accuracy reported is comparable to event extraction in other domains, but the combined evaluation of accuracy and human effort in event extraction has not been reported elsewhere to my knowledge. The event extraction method developed and its application in the Venturi demo is thus a contribution to achieving 4.2.

**Extraction of sleep patterns from smartphone events.** While Paper 1 and Paper 2 focused on using user-generated data with intentionally semantic content, data generated through user behavior can call also be used as to derive activity information that is relevant to social context. In Paper 6, we propose a Bayesian model for extracting sleep patterns from smartphone events. The method can identify individuals' daily sleep periods and their evolution over time and provides an estimation of the probability of sleep and wake transitions.

The model is fitted to more than 400 participants from two different datasets, and we verify the results against ground truth from dedicated armband sleep trackers. We show that the model can produce reliable sleep estimates with an accuracy of 0.89, both at the individual and at the collective level. The Bayesian model can quantify uncertainty and encode prior knowledge about sleep patterns. Compared with existing smartphone-based systems, our method only requires screen on/off events and is therefore much less intrusive in terms of privacy and more battery-efficient.

## 5.2 Research Objective RO2: Assess the feasibility of physical proximity sensing for social context applications.

To assess the feasibility of physical proximity sensing as a mean of allowing applications to use social context as high-level context information an artifact called Proximates was developed. I evaluated this artifact by three evaluation methods from design science:

1. An architectural analysis of Proximates;

2. A case study of an application artifact, Memorit, using Proximates;
3. A field study, in which multiple applications were built using Proximates.

**Architectural analysis**

Proximates, a social context engine software for Android mobile phones is introduced in Paper 3. The paper shows how Proximates can be used to derive social context from proximity sensing in combination with online social networks like Facebook, as well as call and messaging social networks. In an architectural analysis, it also shows how the Proximates architecture satisfies the requirements on mobile social applications defined by Karam and Mohamed (2012):

- Simplification of development process;
- Energy efficiency;
- Privacy;
- Scalability and distributed architecture;
- Heterogeneity and dynamicity of mobile environments.

The analysis also shows how Proximates fulfills the requirements on social context modeling defined by Tran et al. (2009), regarding the explicit capture of relationships, relationship management, and externalization of social context management.

Additionally, during the design and evaluation iteration cycles in the case study, I found that the requirements above needed to be complemented with additional requirements that are not application specific, but needs to be specified for any applications that are aware of social context:

- Latency for detecting context switch;
- Robustness of context detection;
- Accuracy of context detection.

Furthermore, Paper 3 discusses why proximity sensing is preferable to location sensing as proxies for physical social interactions, focusing on the efficiency, latency, robustness, accuracy, and privacy aspects.

The focus in these studies was on *social relations* the context information category in Zimmerman's model, through proximity sensing and acquisition of relationships from external social services, such as Facebook and LinkedIn.

### Case study: Memorit – A reminder application

Proximates was used to prototype several application artifacts, which were then tested in user studies or demonstration to collect feedback. One application, Memorit (first called SmartTodos) was studied closer and for a longer time than the others, going through more iterations of user tests and development, and is thus reported as a case study here. The main objective of the Memorit study was to address RO2 and RO3. RO3 is addressed in Sect. 5.3.

**Artifact, setup, and evaluation.** In Paper 3, I present Memorit (first call SmartTodos), a contextual reminder application used as the artifact in this case study. The main functionality of Memorit is to allow users to set reminders to trigger when conditions on context are met. Users can set a reminder on time and date, location or when they meet a specified friend, i.e., when a specific social context condition occurs (Figure 6).

The app was distributed to a limited group of users, who were recruited using snowballing, i.e., users were recruited by asking users already in the study to recruit more users. Each user received a phone from Sony and downloaded the app from a beta group on Google Play. 175 users participated in the study, running in total 1 year. Not all users participated during the whole time.

**Cold start.** In the study, I faced several challenges. The first one was the *cold start* problem: How can we provide a feature using social context before we have any data about it? To solve this, I used a minimum viable feature approach: The simplest social context that can be detected using proximity sensing is that of being in proximity to one single person. For this, no data needs to be collected and modeled, and I could thus avoid the cold start.

**Informed consent.** The second challenge was that of privacy, and especially that of informed consent. The purpose of the study was to investigate privacy issues formulated in RO3, but I also had to consider the privacy of the participants during the study itself. It is well known that users do not read privacy policies or terms of service, but they still accept such conditions by clicking OK the first time they start an app (Böhme and Köpsell, 2010). Vila et al. (2003) provided a game-theoretic explanation of this phenomena, modeling the collection of user data as an asymmetric privacy information market. They found a unique equilibrium point, where both users and collectors are indifferent to testing claims and respecting privacy due to the high cost of testing privacy policy claims.

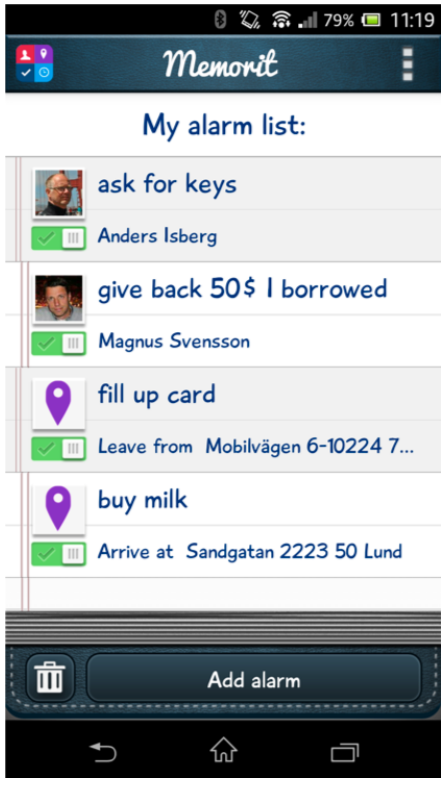



Figure 6: Screenshot from Memorit application showing introduction screen to contacts reminders.

Still, most application developers consider the problem of informed consent to be a legal problem that is dealt with by asking the users to agree to terms of service once only (Luger and Rodden, 2013). Since the users do not read the terms of service, this practice is in violation of user interests and expectations. As Luger and Rodden (2013) point out, informed consent should be considered a social process, where the user is kept informed and consenting, and it is the responsibility of the developer to assure that this is the case.

I formulated the Obvious Data Usage (ODU) principle to address the problem of informed consent in designing and developing applications:

> Any data collected from the user should be reflected in the functionalities of the application collecting the data, in such a way that it is obvious to the user what data is being collected and

how it is being used. (Jonsson, 2012)

This means that if the application developer cannot find a way to make it obvious to the user that the application is collecting a certain piece of data by means of the functionality of the application, then that data should not be collected. ODU is a principle and is hard to test since it is what is obvious to a user may not be so to another. Thus, ODU serves best as an ethical guideline. By applying it to the project at hand, one is forced to reflect on how much the user knows about the data collected and the ethical consequences. ODU is in no way intended to replace privacy laws such as EU General Data Protection Regulation (European Union, 2016).

In this case study, ODU served a double purpose: Not only did it keep the users continuously informed and consenting, but it also made them aware of the proximity sensing. For the privacy survey presented in Sect. 5.3 to be meaningful, the users needed to understand that they were using and being subjected to proximity sensing.

**Power consumption.** Another challenging problem was that of power consumption. Battery life is the most influential factor for consumers when buying smartphones. Thus, phone vendors have strict power consumption requirements on applications they preload. To find a solution that can be deployed in a product, we had to minimize power consumption. While many smartphone users believe that Bluetooth affects power consumption significantly, this is not true. The screen and application CPU consume significantly more power than any other components in a smartphone.

Since Proximates executes as a background service, the application requirements on recency or latency in context sensing determines the power consumption: The lower the latency, the more frequently Proximates needs to execute to scan for Bluetooth. I experimented with various latencies and found that a one-minute frequency was acceptable from a power consumption point of view, while still being able to trigger proximity at an acceptable level. Longer latency made user consider the proximity reminders were not working, and short latency consumed too much power according to users. To reduce power consumption further, I designed a method to control activation of Bluetooth scans (and thus waking the CPU) to only trigger when users were in the same geographical area. This method was filed and granted as a patent application (Jonsson, 2017).

**Usability of the Memorit application.** Paper 3 summarizes an evaluation on task completion of 7 different tasks, including setup. The number of requests for help to complete the task and task completion time were measured. The most complicated task was the setup of the Memorit app,

before it could be used. It includes signing into Facebook, accepting requested permissions, entering a phone number, accepting enabling of Bluetooth and location services if not enabled, and setting Bluetooth visibility timeout to infinity.

The first UI design focused on providing an overview of all these tasks, with little information provided on each task, and allowed the user to complete the setup in any desired order. In the final version after multiple iterations, the UI forced the user to complete the task, step by step, while still indicating overall progress and how many steps were left. The language used in instructions was less technical and more explicit.

Paper 4 reports on the re-evaluation on task completion performed after the final iteration: The task completion time for the two most difficult tasks was reduced by 30% and 70% respectively. Furthermore, the number of requests for help to complete all tasks was reduced by more than 50%. Qualitative user feedback showed that the general user experience had improved significantly. However, setup and configuration were still considered complex, and it was hard to understand how such a simple app could require so much configuration.

**Validation.** I performed a comparative network study to validate that the data collected by Proximates in the Memorit study could be used for social context modeling in terms of relations and that the snowballing recruitment strategy was able to span a representative social network. The analysis is presented in Paper 5.

In this paper, the characteristics of the social network spanned by physical proximity interactions is compared to those of the Reality Mining network (Eagle and Pentland, 2006) and a random Erdös-Renyi network. The Reality Mining network and the Memorit network shared many characteristics, especially the temporal aspects of correlation coefficients. However, over long time frames, the Reality Mining network include many spurious connections (Figure 7) and thus become similar to an Erdös-Renyi network with respect to clustering coefficient and betweenness centrality distributions. The Memorit network is less affected by spurious connections (Figure 8), due to using snowballing recruitment.

**Case study conclusion.** In part, the development, use, and evaluation of Memorit was a *wicked problem*: We iteratively developed an artifact while evaluating it. The requirements changed as we learned from the evaluations with users. The target environment, i.e., Android, changed due to new OS versions and devices with different capabilities being released. The Bluetooth stack in Android was replaced with a different one. Bluetooth Low Energy was partially introduced into Android. Finally, what constitutes

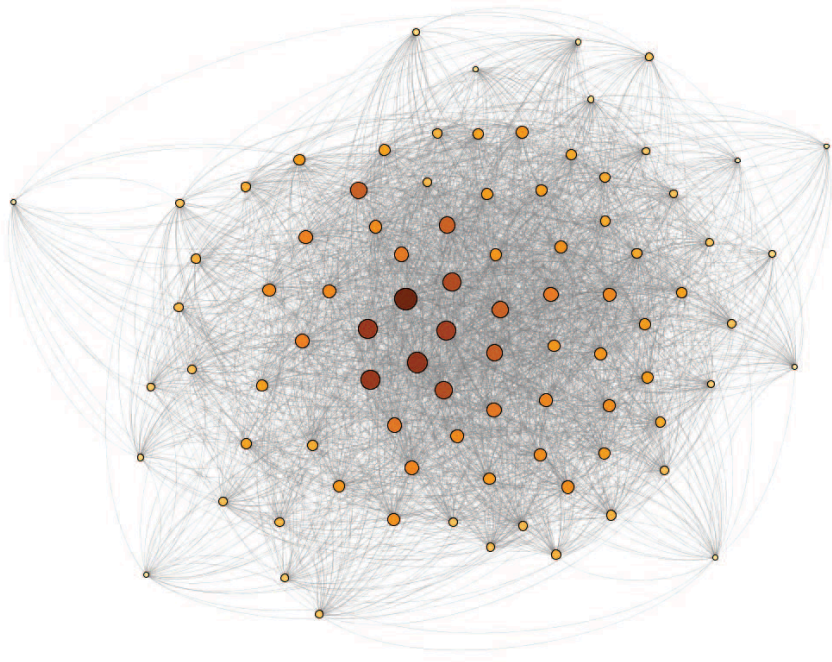


Figure 7: Reality Mining network. Larger node size and darker color indicate higher betweenness

social context that is relevant and usable for applications and users was unclear, and needed to be defined in the process.

These conditions make scientific research challenging. However, if our solution to the problem could not survive such conditions, then it would not be a feasible solution, since these are the conditions in which it needs to work.

The complexity of setup is not an application-specific problem. Any context-aware app needs to request several permissions for accessing relevant data and sensors. In general, this will be hard to explain to the user since it is often not clear: A higher level context may be derived from some lower level sensor, and it is not clear to the user how accessing the sensor contributes to the feature that claims to use it.

This complexity gives phone vendors an advantage over application developers, since they can preload apps with system level permission, without requesting permission from the user. Also, phone vendors can preload apps or services from non-vendor application developers.

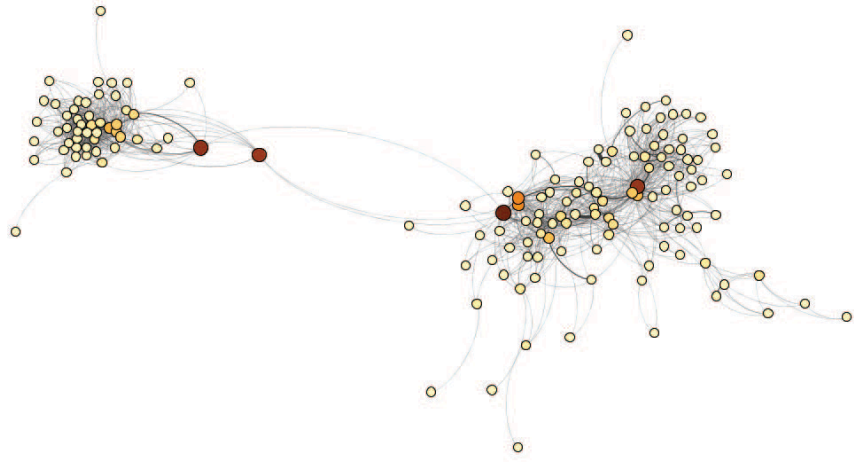


Figure 8: Memorit network. Larger node size and darker color indicate higher betweenness

Phone vendors have to use this advantage with care. Offering context-aware apps without making explicit permission requests to the user may be required from a usability point of view. Vendors may also be the only ones that are trusted enough to be allowed to do this. If they do so, they must also take care to follow ODU and provide value to the user that corresponds to the value of the data they trust the vendors with.

### Field study: Proximates

Three more applications were developed to explore the space of possible proximity sensing applications: Social Photo Frame, Meets, and Stories This exploration also allowed for testing the feasibility and limitations of proximity sensing and Proximates as an implementation.

**Social Photo Frame.** Old photos are great conversation starters. We especially like to see ourselves in photos taken by others and to reminisce about memories they trigger. A digital photo frame application was developed to serve as a conversation starter at home or other private social environments. The application shows old photos figuring the people who are in front of the frame, preferably together. The application queries Facebook for photos that contain people who are also in proximity to the frame. Proximity is detected using Proximates. Proximates can be configured to

map Bluetooth IDs to Facebook IDs and thus used in queries for photos tagged with Facebook IDs.

The application selects photos by maximizing the intersection of people in proximity and tagged in photos. They are then shown in an automatic slideshow.

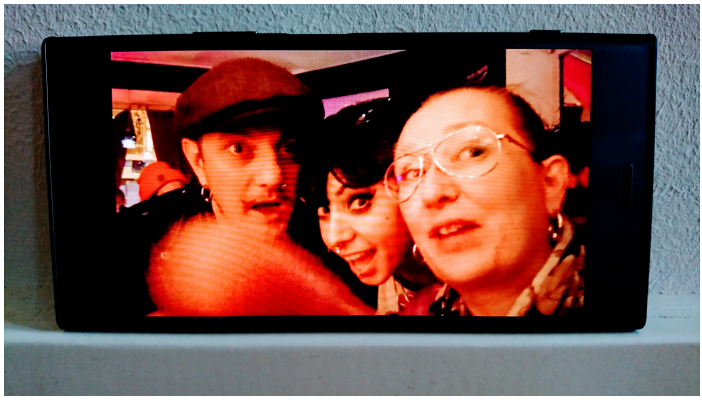


Figure 9: Photo of the Social Photo Frame application

Since most photos are not publicly available on Facebook and need to be accessed on behalf of a specific user, we set this user to be the owner of the frame. To include a person in the query, he needed to be in proximity to the frame, but also already be a Facebook friend of the owner.

Social Photo Frame was demonstrated on three occasions, to collect user feedback. From the demonstrations two important lessons were learned:

- First, externalizing management of relations (to Facebook in this case) allows for reuse of established relations across different applications. However, to support ad hoc use cases such as this, there needs to be a mechanism to establish relationships in a way similar to how humans do it in social situations. As humans, we can interpret thousand of social signals and make immediate interpretations of social order and trust decisions on them. Our relations with other people are not single-valued friendship attributes like on Facebook.

- Second, it is difficult to demonstrate and thus evaluate, social context-aware applications in an artificial environment, such as a conference or a lab. With Social Photo Frame, this meant that all that the people in the audience in the lab saw was a photo of people they did not know. Just like you as a reader can see the application in Figure 9, but that does not tell you anything about how the app works or what

it does. Social Photo Frame worked very well when tested at home with friends since the required relationships were already established. Also, there were in general plenty of Facebook photos to query that made sense.

This difficulty in demonstrating is not only due to the lack of ad hoc relationship creation mechanisms but also due to the wealth of context. For demonstrations of non-context features, some parts are often faked or simulated, but that is not a problem since we can imagine ourselves in the setting. With context features, however, this does not seem to work. One can hypothesize that this is due to the richness and the extensive amount information in what we perceive as context. We are just not able to substitute enough content of our perceived context with the imagined so that it makes any sense to us.

A patent for Social Photo Frame, describing the methods of detecting proximate friends and the retrieval of photos, was filed and granted (Jonsson et al., 2016)

**Meets – A meeting support application.** To further investigate the feasibility of using Proximates to detect groups of people rather than just single individuals, I prototyped an application called *Meets*. Meets helps the user take and communicate notes from business meetings. One feature of Meets was to detect the presence of people in the meeting, i.e., its attendees. The user taking notes could compare the list of invitees and detected attendees and manually correct any errors.

To detect the identity of the owner of a present device, the Proximates (Jonsson and Nugues, 2013) middleware was used. Proximates maps Bluetooth MAC IDs to social identities, for example Facebook. We added support for LinkedIn in the Meets application, since it is used more often in business settings than Facebook.

Paper 7 describes the local filtering method we used in the application. The $k$-nearest neighbor method also described in the same paper achieved slightly better filtering F-score (82%) than using local threshold-based filtering (80%), but required global computation over all data and thus a server. The global method was not implemented in the Meets prototype.

Two lessons were learned:

- First, there is an application-dependent difference in the cost of making mistakes when making autonomous decisions based on context. In Social Photo Frame, there was a notion of an optimal photo to show, but if the application showed a slightly less optimal picture once in a while, no one would notice. The cost of making mistakes was low. In Meets, this cost was higher than in Social Photo Frame. In Mem-

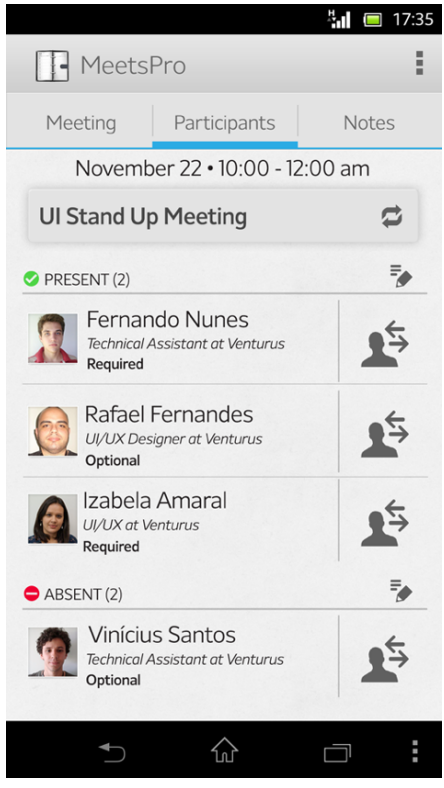



Figure 10: Meets application UI.

orit, it was even higher since a missed reminder could have severe consequences for the user.

- Second, the threshold value found in Paper 7 was too brittle to use in practice. The range of optimal RSSI values was very small and very close to pessimal range (Figure 11). It was not possible to find a value that worked well across environments, devices and group sizes. The $k$-nearest neighbor method is likely to work better since it does not rely on a specific threshold. The brittleness appeared in the application as false positives or false positives, i.e., users not present could sometimes be detected, and users present not. This brittleness was again alleviated in the UI by showing a list of all attendees and whether they were invited and attended, rather than just the detected attendees.

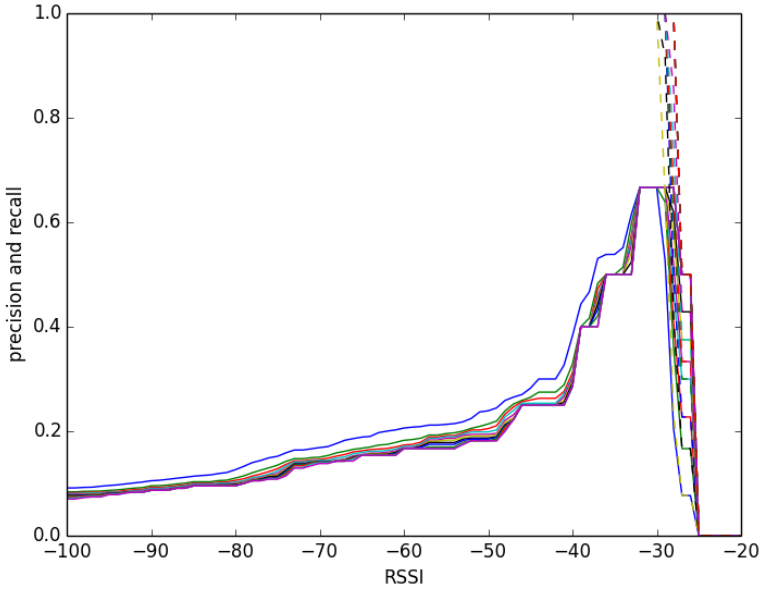



Figure 11: Precision (lines) and recall (dashes) group affiliation classification for different RSSI values

This work resulted in the filing of two patent applications that were granted. One on how to merge calendar and proximity data to reduce uncertainty in data from these sources (Jonsson, 2014c), and one on detection of visitors (Jonsson, 2014b).

**Stories – A self-tracking application.** *Self-tracking* is the tracking of various aspects of one's own life by technological means. There is a multitude of self-tracking applications available that mainly utilize smartphone sensors, sometimes complemented by sensors from wristbands or other wearable devices. None of these applications collect data about who the user meets physically. Some collect data about venues checked into and compare it across users to discover who of a user's friends were at the same place, at the same time. However, check-ins do not necessarily capture physical social interactions since a venue may be large.

We developed a self-tracking application that visualized the places a user has visited, movement activities (walking, sitting, running, etc.), people the user met, and other events (Figure 12).

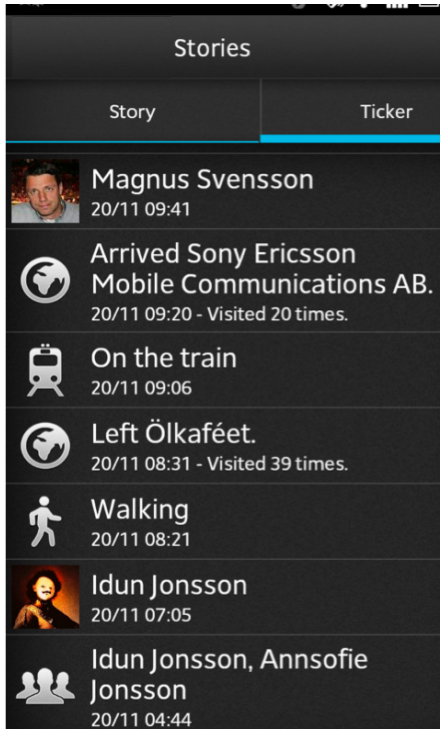



Figure 12: Screenshot from Stories application showing a captured feed of automatically captured activities, events, locations visited, and people met.

To collect feedback about the application, a small qualitative user study was performed. In general, the users found that being able to see what people they had met was a very innovative feature, not available in other self-tracking applications. The applications initially only showed summarized events when something unusual occurred, for example when visiting a new location. Several users were surprised when these events were shown, and thought it somewhat scary from a privacy perspective. We thus applied ODU and decided that we needed to be more transparent about the data collection in the app. We introduced the ticker view, seen in Figure 12, which user thought much cleared and became the main feature of the application. Some users commented that the ticker could be used as a memory prosthetic for associative memory search.

In addition to the privacy aspect, we learned that the completeness in context information being visualized in the ticker UI made it much easier to demonstrate than Social Photo Frame. I think the reason is that we are

visualizing the context information rather than using it as a trigger or cue for some other functionality. In this sense, Stories is not just a context-aware application, but also a context-visualizing application. Potentially, this could mean that context information should be visualized in context-aware applications in general, to make it easier for users to understand and accept.

#### RO2 Conclusion

Proximates made the development of the applications used in the study easy by abstracting away from the low-level sensor APIs in a single configurable software component. It fulfills the architectural requirements for applications that need to use proximity sensing for social context awareness. Proximity sensing is feasible for applications as long as the application developer consider the following:

- Any context sensing app requires complex configuration due to the many permissions needed, and proximity sensing adds yet more permissions. This can be solved by application deployment by a trusted vendor that does not need to requests permissions.
- Proximity sensing is error-prone like all sensing. Thus the cost of making mistakes needs to be considered and potentially solved through UI adaptation, giving the user the chance to detect and manage mistakes.
- Make an appropriate tradeoff between latency and energy consumption requirements. Memorit pushed the boundaries of what is acceptable to users, but future phone platforms will allow low power background proximity sensing just like they have made it possible with Wifi over the past few years.

## 5.3 Research Objective RO3: Understand users attitudes on privacy with respect to proximity data.

By its nature, context information is often sensitive from a privacy point of view. Users have over time become accustomed to having sensed and contextual data collected and used through services such as Facebook collecting location data through GPS on users' smartphones. User attitudes on privacy with respect to location data has been extensively studied, but not attitudes towards proximity sensing. During the Memorit case study, I conducted a survey with the subjects to investigate this. As we have seen before, imagining contexts of other people and contextual features in applications can be hard. Therefore, this survey could only have been done with subjects exposed to proximity sensing applications such as Memorit to be of

any value. Surveys using hypothetical questions about context, especially for contexts not commonly used in applications, is likely to be misleading since subjects would not have mental model informed by the experience of such features.

The results presented in Paper 8 conclude that proximity sensing is not considered more sensitive than location sensing by users. Also, as long as users are in control of who can sense the data, they accept being discoverable by others through proximity sensing.

## 5.4 Research Objective RO4: Estimate how sensitive context data is with respect to privacy, in terms of risk of user re-identification.

de Montjoye et al. (2013) showed that four spatiotemporal points in a location trace dataset are enough to uniquely identify 95% of the individuals. This was highly relevant not only in terms of privacy consequences for context-aware applications but also because location data of many smartphone users is being collected and sold to advertisers through data brokers.

In Paper 3, I argue that proximity data can be less intrusive than location data while still fulfilling the requirements for social context-aware applications. So what about the other categories of context information besides *relations* and *location*? Are they as sensitive as location data with respect to re-identification?

Using the same uniqueness framework as de Montjoye et al. (2013), applied to smartphone application usage data, we calculated the uniqueness of users in a dataset containing 3.5 million users. The usage data only contained data about whether an application had been used during a month or not. It did not include any information about timestamps, duration, number of starts, etc. Each user was just a binary vector, where each position represented a specific app. Still, it turned out that users are highly unique in this data. Given four applications we could uniquely re-identify 91% of the users using a simple heuristic strategy based on selecting the least popular applications. We also showed that uniqueness change over time. During summer months, users become more unique. Also, users change behavior over time; their application usage fingerprint drifts over time with a roughly constant rate.

This is important not only for context-aware applications to consider, but also because application usage data is being collected, sold, and used for advertising on a massive scale. Data brokers collect this data from billions of users together with browser cookies, location data, device identifiers, and other profiling data in so-called Data Management Platforms, and then sell access to the data.

We surveyed the most popular applications (>100,000 downloads) on Google Play that collect application usage data. 25 of these 40 did not have any functionality that would obviously need application usage access to implement their functionality. We guess that this data is being collected to be sold to data brokers.

The results in Paper 9 have practical consequences for anonymization. The most common form of anonymization, $k$-anonymity, is defined in terms of indistinguishability:

> A $k$-anonymized dataset has the property that each record is indistinguishable from at least $k-1$ others.

To perform the anonymization, suppression and generalization are applied to the data. This reduces uniqueness in a dataset by reducing the resolution of data (generalization) or by deleting data completely (suppression). Both suppression and generalization are destructive methods, which means that information and thus the value of the data is lost to a degree when applying them. To trade-off degree of anonymity vs utility value of data, we must first quantify the utility of the data. As an example, if we want to share some application usage data for research purposes, how do we define the utility and how do we select $k$? We take a small sample of the dataset in Paper 9 of 93,000 users and define the utility as the number of users left in the dataset after anonymization. Since each user in this set is just a binary vector, there is no way to perform generalization to anonymize. Instead, we use suppression by deleting users who are distinguishable from more than $k-1$ other users.

To investigate the effect of $k$ and the number of applications on the number of user remaining after anonymization, we order the applications by decreasing popularity and perform anonymization by suppression. The results are plotted in Figure 13. In the figure, we can see that after including more than 10 apps in the binary vector, the number of remaining users drop quickly until about 25 applications, when the drop flattens out. The difference between different values of $k$ is surprisingly small (except for $k = 1$, which is expected since then users don't need to be distinguishable from anyone else).

Since context information is high dimensional by nature, re-identification is possible given enough resources, and anonymization of the data is very hard if any utility is to be retained. Thus, in conclusion, we should treat all context data as personal data.

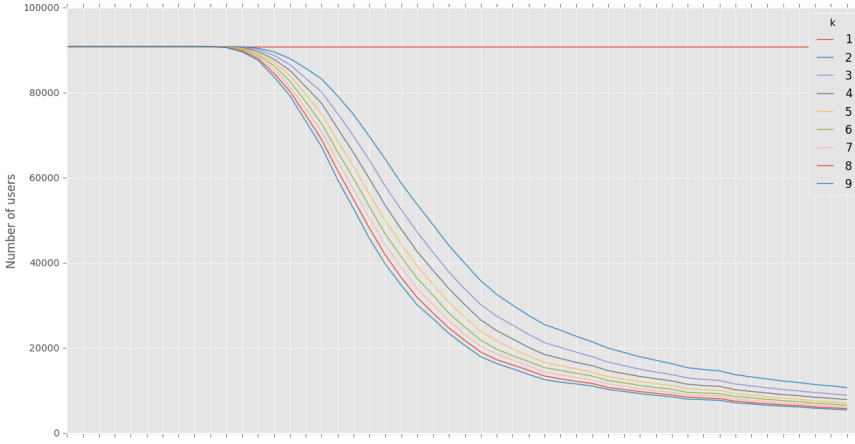



Figure 13: Number of users remaining in an app usage dataset after anonymization using suppression, with incremental inclusion of apps from left to right. Each tick on the x-axis represent the number of the most popular apps included.

# 6 Conclusion and discussion

The overall goal of the work presented in this thesis is to take steps towards making it possible for context-aware applications to make use of social context. To this end, I have conducted research in three areas:

- Acquisition of social context information from user-generated content;
- Sensing of physical proximity;
- Privacy aspect of social context information.

The publications in each area build on empirical research using implementations of methods or application artifacts evaluated on collected data, in case studies and field studies. In terms of the research objectives, I have made the following contributions:

**RO1: Develop methods for the acquisition of social context information from user-generated data (acquired context).**

- A proposed method for extracting event information from social communication, specifically SMS;
- A proposed method for extracting event information from hyper-local sources of information, i.e., web pages;
- A described model of sleep activity based on phone usage.

**RO2: Assess the feasibility of physical proximity sensing for social context applications.**

- A software component for social context sensing, applied in building four different applications and used as artifacts in an architectural analysis, a case study, and a field study.

**RO3: Understand users' attitudes on privacy with respect to proximity data.**

- An investigation of user attitudes towards privacy aspects of proximity sensing.

**RO4: Estimate how sensitive context data is, with respect to privacy, in terms of risk of user re-identification.**

- Through modeling uniqueness, I have shown how sensitive context information is in terms of re-identification risk.

These contributions are summarized in Table 1.

| RO | Papers | Artifacts | Dourish RQs | Evaluation |
|---|---|---|---|---|
| RO1 | 1, 2, 6 | Method impl., app | 2, 3, 6 | Method performance, user study |
| RO2 | 3, 4, 5, 7 | Proximates, Memorit, Meets, Social Photo Frame, Stories | 2, 3, 5, 6 | Architectural analysis, Case study, Field study |
| RO3 | 8 | Memorit | 4, 5 | Survey |
| RO4 | 9 | | 5 | Data analysis |

Table 1: Research objectives, artifacts, and papers

My research objectives have been inspired by the current state of research in context-awareness and ubiquitous computing. The theoretical foundation behind context-awareness within ubiquitous computing draws on logic to provide the basis for representation, reasoning and management, and is a positivist approach. In the human-computer interaction tradition, the theoretical foundation is more philosophical and based in phenomenology.

In his seminal paper "What we talk about when we talk about context", Dourish (2004) proposes a different view on context than the positivist view that for example Zimmermann's model and STIPI (Schuster et al., 2013) takes. Instead of the positivist view of context as a representation and modeling problem, Dourish proposes a phenomenological approach, viewing context as an interaction problem. Dourish thinks of social context

as an emergent relational property that does not lend itself to meaningful modeling. In this view, social context and activity can not be separated.

I agree with Dourish that it is not clear that social context can be meaningfully represented in the form of explicit ontologies. There may be too many subtle nuances that are too complex to be captured in ontologies. Even if we cannot represent these subtleties, that does not mean we cannot model context as an achieved and maintained mutual inter-subjective experience. Using machine learning, embodied agents can model context. The models have an internal representation, but they are complex and rarely offer explainability. The question whether this is representational and positivist or embodied and phenomenological is mainly philosophical since representation learning is a common feature of machine learning methods today.

Dourish (2004) formulated general research questions for context awareness. These have been important in the process of formulating my research objectives. His research questions are:

1. "What role does context play in our everyday experience?"
2. "How can this be extended to a technological domain?"
3. "What can the computation really do for us?"
4. "How can we interact with [it] as an invisible presence and yet maintain adequate control?"
5. "How can we feel both served and safe?"
6. "What are feasible context cues of social context?"
7. "What is the appropriate representation of social context to allow for action/reflection/adaptation?"

These are general research questions regarding context, not only social context. I have not addressed them all in this thesis. Yet, some of my research objectives are relevant and make contributions to Dourish more general questions. Each of my research objectives is mapped to how they contribute to each of Dourish research questions in Table 1.

## 6.1 Discussion

**Embodiment and externalization of relations.** With RO2, I have explored a class of use cases, where we need our technology to represent us in the sense that we become discoverable by technology representing others. To give an example, we can think of clothes as a technology that people who know a person can use to visually identify him at a distance, or he can use

it to express identity as belonging to a certain social class or community, allowing people he never met to ascertain some aspect of his identity.

The difference in the case of clothing is that we don't necessarily need technology for the discovery part. Clothes is a good example of this class, where the technology works as an extension of us and represent us. We often see clothes as an extension of ourselves, doing its job without need for interaction (except when the zipper gets stuck). Ihde (1990) describes this relation to technology as being *embodied*. I call technology that we can have an embodied relation to, and that can represent its user an *embodied social agent*.

To make the embodied social agent use cases possible, we want technology we carry, such as smartphones, to represent us to the physical and digital world by bridging these. However, today this is rarely possible in an embodied fashion. Instead, we are often forced into an *alterity* relationship, where the device is a separate entity we need to interact with, even when this is not the ideal relationship for the interaction being designed.

As an example of an alterity relation, think of the case of exchanging electronic business cards. If we want to transmit it locally to someone in a meeting room, we first have to ask them which device is theirs to find it in the list of discoverable devices, rather than send it to them as persons, transparently represented by their devices. Thus, for our use cases, we need a mechanism for embodied agents to establish interpersonal relationships, potentially through embodied agents of the counterpart. This mechanism needs to support the ad hoc formation of trust relations with all the complex social facets and nuances that constitute human trust establishment.

Externalized management of relations allows us to separate management of trust relations from the application, and reuse the relations across applications. It can abstract away from how trust relations are established. For example, Facebook APIs can be used to invite friends (already established relations) to use an application, independent of application. It does not allow users to establish relationships through any other mean than through the Facebook application. Also, the relations models are far too simple to capture human trust relations well enough to for embodied agents to be able to represent us. Existing systems thus cannot be used for embodied social agency use cases. It is not clear that even existing proposed models for modeling relationships by means of ontologies can capture the complexity.

An alternative approach, using machine learning to model relations, can potentially be used for embodied representation use cases. They may not be able to provide human-readable representation of the relations or a representation suitable for reasoning using classical methods, but they can still be used to implement solutions for the use cases.

This problem reflects the classic dispute between embodied cognition

and representational cognition (Brooks, 1991). As interesting as this philosophical debate is, we are interested in actual representation so we can use it in applications. The word embodied, as in embodied cognition, is used in a different sense than by Ihde meaning "cognition in a body", while Ihde is referring to a relationship between a user and technology. Establishing, representing, managing, and using relations for embodied social agency are all topics for future research.

Lindblom (2015) has studied embodied social interaction and its implications for embodied cognition in artificial intelligence. She has proposed an integrated cognitive science of human-computer action (HCI) along these lines of social embodied cognition and Dourish (2004). Her interest is not in solving the problem of the embodied social agency use case class, but rather within embodied social interaction and cognition in general and its implications for HCI.

This line of research could be a very promising approach for studying the embodied social agency problem, unifying the two meanings of the word embodied.

## 6.2 Closing

The work presented here has been a journey full of experimentation and learning. As with any interesting research topic, it has also led to new questions. Would I be given a chance to continue researching these topics, I would further investigate these questions:

- What technologies can support general embodied social agency of our physical and digital selves in both the physical and digital realms?
- What technology can support the ad hoc establishment of trust relations in the physical and digital realms?
- How can we meaningfully represent interpersonal and inter-agent trust relations in the physical and digital realms so that we can externalize and reuse them?

# Bibliography

Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer.

af Segerstad, Y. H. (2002). *Use and adaptation of written language to the conditions of computer-mediated communication*.

Aharony, N., Pan, W., Ip, C., Khayal, I., and Pentland, A. (2011). Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 7(6):643–659.

Atzmueller, M., Ernst, A., Krebs, F., Scholz, C., and Stumme, G. (2014). On the evolution of social groups during coffee breaks. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 631–636. ACM.

Atzmueller, M. and Hilgenberg, K. (2013). Towards capturing social interactions with sdcf: An extensible framework for mobile sensing and ubiquitous data collection. In *Proceedings of the 4th International Workshop on Modeling Social Media*, page 6. ACM.

Bardram, J. E. (2005). The java context awareness framework (jcaf)–a service infrastructure and programming framework for context-aware applications. In *International Conference on Pervasive Computing*, pages 98–115. Springer.

Barrat, A., Cattuto, C., Colizza, V., Pinton, J.-F., Broeck, W. V. d., and Vespignani, A. (2008). High resolution dynamical mapping of social interactions with active rfid. *arXiv preprint arXiv:0811.4170*.

Battestini, A., Setlur, V., and Sohn, T. (2010). A large scale study of text-messaging use. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pages 229–238. ACM.

Bell, S., McDiarmid, A., and Irvine, J. (2011). Nodobo: Mobile phone as a software sensor for social network research. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180.

Böhme, R. and Köpsell, S. (2010). Trained to accept?: A field experiment on consent dialogs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2403–2406. ACM.

Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A., and Tanca, L. (2007). A data-oriented survey of context models. *ACM Sigmod Record*, 36(4):19–26.

Brooks, R. A. (1991). Intelligence without representation. *Artificial intelligence*, 47(1-3):139–159.

de Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013). Unique in the Crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7.

Dey, A. K., Abowd, G. D., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166.

Do, T. M. T., Blom, J., and Gatica-Perez, D. (2011). Smartphone usage in the wild: a large-scale analysis of applications and context. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 353–360. ACM.

Dourish, P. (2004). *Where the Action is*. MIT press, Cambridge, Mass.

Eagle, N. and Pentland, A. (2005). Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 4(2):28–34.

Eagle, N. and Pentland, A. S. (2006). Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268.

Eagle, N., Pentland, A. S., and Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. *Proceedings of the national academy of sciences*, 106(36):15274–15278.

European Union (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119:1–88.

Ferreira, D., Kostakos, V., and Dey, A. K. (2015). Aware: mobile context instrumentation framework. *Frontiers in ICT*, 2:6.

Génois, M., Vestergaard, C. L., Fournet, J., Panisson, A., Bonmarin, I., and Barrat, A. (2015). Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science*, 3(3):326–347.

Giulio and Chippendale (2014). D6.7 use cases integrated on venturi integrated platform. Technical report.

Google (2015). `https://developers.google.com/nearby/`.

Google (2016). `https://developers.google.com/awareness/`.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1):75–105.

Ihde, D. (1990). *Technology and the lifeworld: From garden to earth*. Number 560. Indiana University Press.

Iwatani, Y. (1998). `https://www.wired.com/1998/06/love-japanese-style/`.

Jiang, H., Wang, X., and Tian, J. (2010). Second-order hmm for event extraction from short message. In *International Conference on Application of Natural Language to Information Systems*, pages 149–156. Springer.

Jonsson, H. (2012). The data chicken and egg problem. In *Workshop on Informing Future Design via Large-Scale Research Methods and Big Data*, pages 7–10.

Jonsson, H. (2014a). User-based semantic metadata for text messages. US Patent 8,849,930.

Jonsson, H. (2014b). Visitor detector. US Patent 8,787,886.

Jonsson, H. (2017). Power efficient proximity detection.

Jonsson, H., KRISTENSSON, A., and Isberg, A. (2016). Adaptive media object reproduction based on social context. US Patent 9,313,318.

Jonsson, H. and Nugues, P. (2013). Proximates–a social context engine. In *Evolving Ambient Intelligence*, pages 230–239. Springer International Publishing.

Jonsson, H. L. E. (2013). Text enhancement. US Patent 8,588,825.

Jonsson, H. L. E. (2014c). Verifying calendar information through proximate device detection. US Patent 8,737,950.

Karam, A. and Mohamed, N. (2012). Middleware for mobile social networks: A survey. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 1482–1490. IEEE.

Khalil, A. and Connelly, K. (2005). Improving cell phone awareness by using calendar information. In *IFIP Conference on Human-Computer Interaction*, pages 588–600. Springer.

Khattak, A. M., Akbar, N., Aazam, M., Ali, T., Khan, A. M., Jeon, S., Hwang, M., and Lee, S. (2014). Context representation and fusion: advancements and opportunities. *Sensors*, 14(6):9628–9668.

Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., and Laurila, J. (2010). Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*.

Lazar, J., Jones, A., Hackley, M., and Shneiderman, B. (2005). Severity and impact of computer user frustration: A comparison of student and workplace users. *Interacting with Computers*, 18(2):187–207.

Lindblom, J. (2015). Embodiment and social interaction. In *Embodied Social Cognition*, pages 115–159. Springer.

Lu, H., Yang, J., Liu, Z., Lane, N. D., Choudhury, T., and Campbell, A. T. (2010). The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 71–84. ACM.

Luger, E. and Rodden, T. (2013). Terms of agreement: Rethinking consent for pervasive computing. *Interacting with Computers*, 25(3):229–241.

Miluzzo, E., Lane, N. D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S. B., Zheng, X., and Campbell, A. T. (2008). Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350. ACM.

Miluzzo, E., Papandrea, M., Lane, N. D., Lu, H., and Campbell, A. T. (2010). Pocket, bag, hand, etc.-automatically detecting phone context through discovery. *Proc. PhoneSense 2010*, pages 21–25.

Montanari, A., Nawaz, S., Mascolo, C., and Sailer, K. (2017). A study of bluetooth low energy performance for human proximity detection in

the workplace. In *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*, pages 90–99. IEEE.

Moran, T. P. and Dourish, P. (2001). Introduction to this special issue on context-aware computing. *Human–Computer Interaction*, 16(2-4):87–95.

Mostarda (2014). D5.5 prototype software and extended report on 3d social data mining. Technical report.

Olguín, D. O., Waber, B. N., Kim, T., Mohan, A., Ara, K., and Pentland, A. (2009). Sensible organizations: Technology and methodology for automatically measuring organizational behavior. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):43–55.

Opoku-Boateng, G. A. (2015). User frustration in hit interfaces: Exploring past hci research for a better understanding of clinicians' experiences. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1008. American Medical Informatics Association.

Raento, M., Oulasvirta, A., and Eagle, N. (2009). Smartphones: An emerging tool for social scientists. *Sociological methods & research*, 37(3):426–454.

Schuster, D., Rosi, A., Mamei, M., Springer, T., Endler, M., and Zambonelli, F. (2013). Pervasive social context: Taxonomy and survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):46.

Stopczynski, A., Larsen, J. E., Lehmann, S., Dynowski, L., and Fuentes, M. (2013). Participatory bluetooth sensing: A method for acquiring spatio-temporal data about participant mobility and interactions at large scale events. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 242–247. IEEE.

Szomszor, M., Kostkova, P., Cattuto, C., Van den Broeck, W., Barrat, A., and Alani, H. (2011). Providing enhanced social interaction services for industry exhibitors at large medical conferences. In *Developments in E-systems Engineering (DeSE), 2011*, pages 42–45. IEEE.

Tran, M. H., Han, J., and Colman, A. (2009). Social context: Supporting interaction awareness in ubiquitous environments. In *Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous' 09. 6th Annual International*, pages 1–10. IEEE.

Van den Broeck, W., Cattuto, C., Barrat, A., Szomszor, M., Correndo, G., and Alani, H. (2010). The live social semantics application: a platform for integrating face-to-face presence with on-line social networking.

In *First International Workshop on Communication, Collaboration and Social Networking in Pervasive Computing Environments (PerCol'10)*.

Vila, T., Greenstadt, R., and Molnar, D. (2003). Why we can't be bothered to read privacy policies models of privacy economics as a lemons market. In *Proceedings of the 5th international conference on Electronic commerce*, pages 403–407. ACM.

Wu, L., Waber, B., Aral, S., Brynjolfsson, E., and Pentland, A. (2008). Mining face-to-face interaction networks using sociometric badges: Predicting productivity in an it configuration task.

YAC (2017). `http://yacworld.com/`.

Zimmermann, A., Lorenz, A., and Oppermann, R. (2007). An operational definition of context. *Context*, 7:558–571.

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 27 (2011) 178 – 187

Procedia
Social and Behavioral Sciences

Pacific Association for Computational Linguistics (PACLING 2011)

# Named Entity Recognition for Short Text Messages

Tobias Ek[a]*, Camilla Kirkegaard[a], Håkan Jonsson[b], Pierre Nugues[a]

[a]*Lund University, Department of Computer science, Box 118, S-221 00 Lund, Sweden*
[b]*Sony Ericsson, Nya vattentornet, S-221 88 Lund, Sweden*

**Abstract**

This paper describes a *named entity recognition* (NER) system for short text messages (SMS) running on a mobile platform. Most NER systems deal with text that is structured, formal, well written, with a good grammatical structure, and few spelling errors. SMS text messages lack these qualities and have instead a short-handed and mixed language studded with emoticons, which makes NER a challenge on this kind of material.
We implemented a system that recognizes named entities from SMSes written in Swedish and that runs on an Android cellular telephone. The entities extracted are *locations*, *names*, *dates*, *times*, and *telephone numbers* with the idea that extraction of these entities could be utilized by other applications running on the telephone. We started from a regular expression implementation that we complemented with classifiers using logistic regression. We optimized the recognition so that the incoming text messages could be processed on the telephone with a fast response time. We reached an F-score of 86 for strict matches and 89 for partial matches.


*Keywords:* Named entity recognition; Short text messages; SMS; Information extraction; Ensemble systems;

## 1. Introduction

Named entity recognition (NER) from short text messages (SMS) on handsets has until now been constrained by computing power and memory. Current implementations only detect telephone numbers or hyperlinks using regular expressions. Most implementations are optimized for a high recall and usually match any number as a telephone number.

Telephone numbers and hyperlinks are not the only valuable named entities within SMSes. Locations, names, telephone numbers, dates, and times are all potentially important information that the user may

* Corresponding author. Tel. 0046705973300
*E-mail address*: tobias.ek@telia.com.


doi:10.1016/j.sbspro.2011.10.596

want to use in other applications. For example, the user may want to perform a map search on a location name occurring in a SMS. Recognizing these entities is the first step to allow the user to pass the data on to other applications with minimal user interaction.

This paper describes a system that recognizes named entities from SMSes written in Swedish and that runs on an Android cellular telephone. Entities detected are telephone numbers, dates, times, locations, and person names. We gathered a corpus to evaluate the detection accuracy and we compared it with that of existing published systems. Although not exactly comparable, we report results that we believe are on a par with what could be expected from a mid or high-end computer. One of the essential features of our NER system is that we started from a regular expression implementation that we improved with corpus-trained classifiers using logistic regression.

## 2. Previous Work

Little work on named entity recognition in constrained environments has been published. Jiang et al. [7] investigated the use of hidden Markov models (HMM) to extract named entities related to events or activities from SMSes in Chinese. The execution was specifically targeted to be for handsets. While they achieved a lower F-score, the authors could reduce significantly memory consumption. However, they used a SMS corpus of 1,000 messages, which is relatively small compared with sizes common in the field. They combined it with a larger corpus using daily newspaper data.

Polifroni et al. [11] used logistic regression to recognize name, location, date, and time entities from spoken or typed messages. They built a corpus from transcribed utterances and English SMSes from real users in a laboratory setting. They reported F-scores for names and locations reaching 88 on an individual word basis. The end goal is to use the system in a mobile setting for automatic speech recognition, but they do not report on computational or memory resources required of their approach.

Hård af Segerstad [6] provides an extensive analysis of the linguistic characteristics of Swedish SMS texts and usage of SMS in Sweden. She found that SMSes contain unconventional and not yet established abbreviations based on Swedish as well as words from other languages, unconventional or spoken-like spelling, unconventional use of punctuation, and use of non-alphabetical graphical means, e.g. emoticons. For NER, this means that SMS is a particularly hard domain.

## 3. Corpus: Collection and Annotation

We built a corpus using incoming and outgoing messages from 11 participants. Messages were written mainly in Swedish, but sometimes mixed with English and German. This corresponds to a realistic SMS use in an international setting in Sweden. We collected this corpus in parallel with the development of the NER system and we reached a size of about 4,500 text messages consisting of 60,000 tokens.

Such a size seems to be at the lower limits in terms of data quantity needed to have a reliable evaluation [1]. A larger amount of data would of course guarantee a more accurate result, but at the expense of a more costly gathering procedure. Even if the figure of 60,000 tokens seems limited, the gathering task turned out to be surprisingly difficult as many users consider a SMS to be private, if not intimate. Most users were unwilling to share their data unless an option was offered to exclude certain private text messages.

We annotated the corpus with five categories of named entities potentially useful for applications; see Table 1. As markup language, we used the IOB2 format [12] with the tags: B (begin), I (inside), and O (outside). In our corpus, around 90% of the tokens are tagged as outside. Below is an example of a bracketed message:

*Nu åker vi in till [$_{LOC}$ stan]. Ska vi ses [$_{TIM}$ kl 17.15] på [$_{LOC}$ max]? Puss*
'We are driving into [$_{LOC}$ town] now. Should we meet at [$_{TIM}$ 17.15] at [$_{LOC}$ max]? xxx'

Table 1. Named entity types and examples

| Entity | Tag | Examples | Entity | Tag | Examples |
|---|---|---|---|---|---|
| Date | DAT | *10-09-22*, *22/09/10* | Name | PER | *Tobias*, *Torsten Andersson* |
| Time | TIM | *12:34*, *klockan nio* | Location | LOC | *Lund*, *skolan* |
| Telephone no. | PHO | *073-123456*, *+464612345* | | | |

We annotated all the tokens of the tokenized corpus with the five categories we wanted to extract and their corresponding IOB2 tag. Annotation is a time-consuming and costly process that requires a good deal of human effort. As an initial step, we *bootstrapped* this process by applying a set of manually written regular expressions to detect and label the entities. As a second step, we corrected the entity labels by hand to produce the final corpus.

In addition to being concise, regular expressions are very effective in finding numerical tokens such as dates, times, and telephone numbers that appear in recurring patterns even across disparate formatting styles. However, they are insufficient for entities that differentiate too much from these simple patterns.

During the development, we utilized a development set consisting of 2,000 tokens (214 text messages). This set was throughout used for regression testing, feature selection, system comparison, and general error analysis.

## 4. The Initial Regex-based System

We started the NER system with a regex-based implementation. Table 2 shows examples of regexes for numerical expressions. We evaluated this system with a script derived from the one used in CoNLL 2003 [13] and we reached a F-score close to 74 on the development set.

Table 2. Sample of regular expressions used to detect numerical entities

| Regex | Matches |
|---|---|
| `den[ ](([1-3][0-9])\|([1-9]))` | den 23 |
| `kl(ockan)?[ ](\d{1,2}[:.]\d{2})` | kl 13:37 |
| `(2[0-3])(([:.])[0-5]\d)\{1,2\}` | 21.52 |
| `[0-2]\d[0-5][0-9]` | 0845 |
| `((00\|[+])\d{2}[-\s]?([\s]?\d){6,})` | +46-123456 |
| `\d{2,}-\d+` | 08-123456 |
| `[+][+]?\d+[ ]?(\d{1,}\|[ ()]){2,}` | ++45 (404) 354 54 |

The system accuracy was acceptable for number-based tokens, mostly dates, times, and telephone numbers, as long as the entities were more or less well formed. Nonetheless, the regular expressions quickly reach their limits when users are using nonstandard formats. Although, it is possible to continue developing a regex set, any improvement comes with increasing complexity and reduced manageability.

Finding letter-based tokens, like names and locations with regexes limits the system to an already known datasets (i.e. lists) with the exception of suffixes for locations. One could gamble and try

expressions that look at word patterns such as “at the” but such guesswork is better handled by a classifier. There is also an ambiguity with names and locations that is not easily solved; locations, such as restaurants or coffee shops, are often named after a person.

## 5. Named Entity Detection using Classifiers

The architecture of the classifier-based system consists of a pipeline of components; see Figure 1. The training steps are carried out on a desktop computer and the recognition steps on a cellular telephone. The training procedure takes a SMS corpus as input, tokenizes it, and tags each token with its part of speech using tools from the OpenNLP toolkit [10]. Finally, we trained our recognition models using logistic regression from the LIBLINEAR package [4]. The recognition procedure on the cellular telephone uses a similar pipeline except that it utilizes the linear regression models produced in the training phase to mark up the named entities.

### *5.1. Part-of-Speech Tagging*

We used two part-of-speech taggers: Granska [2], a high-performance tagger for Swedish written in C++ for the training phase, and the OpenNLP tagger written in Java to run on the Android platform. We had to use Granska first to annotate our corpus, as OpenNLP has no model for Swedish. We could not use Granska on Android, as it is not written in Java.

We also wanted to control the size of the models due to memory restrictions for the cellular telephone. An application on Android cannot allocate more than 16 MB of memory and with OpenNLP, we can adjust the model size using smaller amounts of training data. The price of a smaller POS tagger is unfortunately that it will be less accurate.

### *5.2. Design of a Feature Set*

We trained the named entity classifier with a set of features that we extracted from the tokens. Finding efficient features can be a never-ending task. We first built a feature superset from sets described by [3] and [5]. We then created our own features. Throughout the development, we carried out regression tests using a forward greedy selection to verify that features had positive impacts on the performance and design an optimal set.

We used a set of about 30 features and Table 3 shows the major ones. As in [8], we extracted these features using a window of three tokens before and after the current token. We used the lexical value and part of speech of the tokens. Most of the features in Table 3 have a self-explanatory name. Some features reflect properties of the current token: digits, letters, prefixes, and suffixes; some consider the part-of-speech tags. The *contain features* use either regular expressions or lookups from lists. Finally, some features model the context by looking at surrounding words.

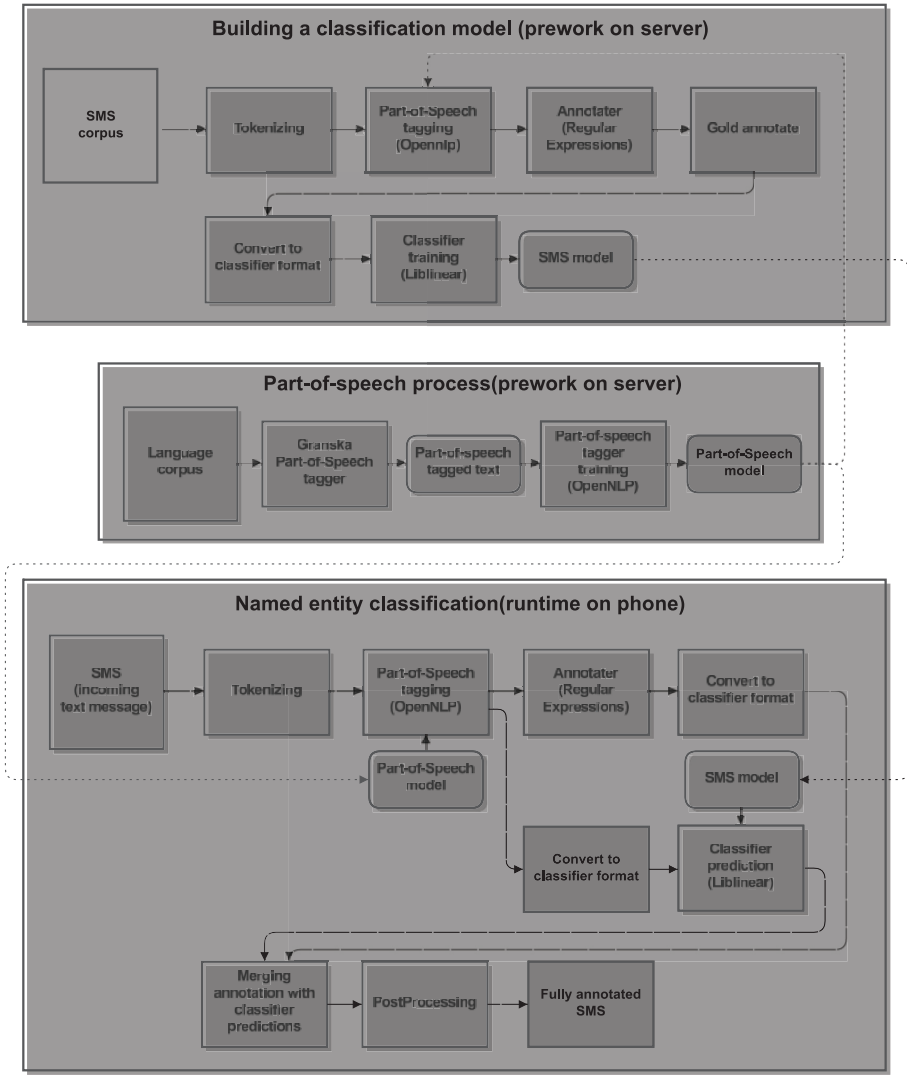



Figure 1 Overview of the major steps and components of the NER system

Table 3. The major features

| Type | Regex | Matches | Type | Regex | Matches |
|---|---|---|---|---|---|
| **Regex** | Contains date | *Augusti, Måndag* | **List** | Contains derived word | |
| | Contains times | *12:00, klockan 11* | | Common word | *bil* 'car', *hus* 'house' |
| | Contains telephone no. | *046-123456* | | Contains common name | *Maria*, *Peter* |
| **POS** | Part-of-speech tags | noun, verb | | Contains derived name | |
| | Verb | *gå* 'walk', *läsa* 'read' | | Contains location | *Stockholm*, *Malmö* |
| | Specific verb | *träffas* 'meet', *äta* 'eat' | | Contains derived location | |
| | Noun | *tåg* 'train', *bok* 'book' | | Frequent unigrams | *södra* 'south' |
| | End of sentence | !.? | | Frequent bigrams | *jag och* 'me and' |
| | Preposition | *till* 'to', *från* 'from' | | Frequent trigrams | *Jag är på* 'I am at' |
| **Other** | Contains digit | 12:34, 1234 | | Frequent POS unigrams | preposition, noun |
| | Contains denominator | , . - ; : / ) ( | | | |
| | Only digits | 1234567890 | | | |
| | All uppercase | SEMC, XML, NLP | | | |
| | Initial uppercase | *Orange* | | | |
| | Initial lowercase | *orange* | | | |
| | Prep + token + end | *till Lund*. 'to Lund.' | | | |
| | Only letters | *utansiffror* 'withoutdigits' | | | |
| | Length of the word | | | | |

### *5.3. Feature Performance*

We analyzed the performance obtained by different categories of features on the development set described in Sect. 3.2. Table 4 shows the recognition results obtained using sets, where all the features belong to one category only. The main conclusions are that:

- Features based on the POS tags produce relatively evenly distributed figures across the named entity categories. However, the recognition performance of the system would be low if it only relied on POS tags.
- Regular expressions have a good performance, especially with numbers, even when being without support from other features. The classifier responds well on the clues given by the regexes and they are easy to combine with other sets of features.
- Lists of names and locations give a boost to locations and names. Lists containing unigrams, bigrams, and trigrams improve further the performance. N-gram lists need a *cut off* value to keep the system robust. Otherwise the system would overreact to hapaxes.
- Other features deal mostly with the different characteristics of a single token. For instance, one feature checks if the token only contains uppercase letters.

## 6. Lists

Using lists improved the performance for both systems. The classifier responded well on features using information based on the lists. There exists a risk to overgrow the lists and introduce noise, not to mention to slow down performance. We tried to avoid building large lists to mitigate the risk of false positive hits.

Selective smaller lists with relevant content should in most cases be the wiser choice. Mikheev et al. [9] have shown that their content is far more important than their size and a small gazetteer list with well-known entries is far more helpful than a large list listing relatively unknown names, which seldom appear in text.

We gathered lists of people names and locations from sources such as *Statistiska centralbyrån* 'Statistics Sweden' (SCB)[†]. From these lists, we pulled first names, last names, and locations. Table 5 shows our static lists.

In addition to the gazetteer information, we also used lists that we derived automatically from the training data. We extracted a list of frequent words occurring in the training corpus. We also used lists of frequent bigrams and trigrams. We built lists of words and parts of speech that precede a *named entity*. We assigned different cutoff frequencies for each list with the aim to keep the lists as small as possible without infringing on performance.

Table 4. F-score of different feature sets. The *All* column shows the results of the complete feature set. The other columns show the results of feature subsets consisting of only one category

| Tag | POS | Regex | List | Other | All |
|---|---|---|---|---|---|
| TIM | 43 | 67 | 43 | 22 | 87 |
| PHO | 43 | 64 | 67 | 72 | 87 |
| DAT | 31 | 72 | 50 | 0 | 94 |
| LOC | 17 | 57 | 48 | 3 | 72 |
| PER | 22 | 65 | 81 | 0 | 87 |
| Mean | 28 | 65 | 56 | 10 | 84 |

Table 5. Gazetteer information obtained from various Swedish sources

| Category | Size | Example | Category | Size | Example |
|---|---|---|---|---|---|
| First names | 200 | *Adam*, *Maria* | International cities | 150 | *New York*, *Tokyo* |
| Last names | 100 | *Svensson*, *Lundberg* | Points of interest | 15 | *systemet*, *torget* |
| Family relation | 14 | *faster*, *mamma* | Months | 32 | *juli*, *dec* |
| National towns | 2,000 | *Stockholm*, *Lund* | Days | 60 | *julafton*, *torsdag* |

## 7. Evaluation

We developed an evaluation program based on the CoNLL 2003 script [13] using the precision, recall, and harmonic mean of them: F-score. As users may accept a partially correct detection, we computed two F-scores:

- **A strict F-score** that uses the entire tag: A tag is counted as correct if both the prefix, *Begin* or *Inside*, and the entity category, TIM, PHO, DAT, etc., are correct.
- **A partial F-score** that sets aside the prefixes, *Begin* and *Inside*, and uses the entity category. It makes no difference between B-LOC and I-LOC, for instance.

The strict F-score reflects then the proportion of tags that are completely correct, while the partial F-score measures recognitions where the user has to adjust the boundaries of the named entities.

[†] http://www.scb.se

### 7.1. Comparison

We compared the performance of our initial regular expression system with that of the classifier-based one using the development set and our evaluation script. The system based on regular expressions reached a strict F-score of 76.76 and the classifier an F-score of 77.73. Both systems found mostly the same occurrences, but there were differences in their performance. Table 6 shows their strengths broken down by category.

Table 6. Classifier vs. regular expressions. The star (*) shows which system performed best with that particular IOB2 tag

| NE | Classifier | Regex | Difference |
|---|---|---|---|
| B-TIM | | * | 2% |
| I-TIM | | * | 36% |
| B-PHO | | * | 5% |
| I-PHO | | * | 15% |
| B-DAT | | * | 9% |
| I-DAT | | * | 14% |
| B-LOC | | * | 4% |
| I-LOC | * | | 30% |
| B-PER | * | | 3% |
| I-PER | * | | 71% |

### 7.2. Reconciling the Output

We used these differences to build the final ensemble recognition algorithm. It uses a voting procedure, where in case of disagreement the system the most efficient in the category wins. The algorithm is based on trial and error, and the combination that gave the best performance:

- If both systems agree on a tag, this tag is selected.
- If one of the tags is O and the other is a named entity tag, the entity tag is selected.
- If the classifier outputs a tag with the PER category, this tag is selected.
- If both tags are named entities and not of the PER category, the regex annotation is chosen.

## 8. Evaluation Setup and Results

We used a 10-fold *cross validation* to estimate the performance of the ensemble system: regular expressions combined with the classifier-based system. We divided the data set into a test set (1/10) and training set (9/10), where we assigned every tenth text message to the test set.

We broke down the F-scores per entity category to get a more detailed picture of the strengths and weaknesses of our system. Table 7 shows the confusion matrix per category. The O tag is excluded from most tables and F-score calculations, since it is not a named entity like the other tags. Table 8 shows the results for the strict and partial matches using cross validation. As final results, we obtained F-scores of 86.44 for the strict matches and 88.85 for the partial matches on the cross validation.

## 9. Comparison with other NER Systems

We compared the performance of our ensemble system with that of other published systems. This area seems to be new for cellular telephones. Most systems we reviewed are larger and run on desktop computers without specific constraints in memory, response time, or processing power. Most systems are also intended to process formal text. Ours had to handle both informal and formal text. In addition, few use Swedish and their corpora were not accessible to us.

Table 9 shows the key figures that compare our system with the works of [7] and [11]. We could not find any NER system for SMS that targets Swedish as language. [11] did not report the size of corpus, other than qualifying it as large. The SMS corpus was created by users in a laboratory setting and is used together with a corpus of transcribed voice notes. [7] used a small SMS corpus for testing and a large newspaper corpus for training. They extracted person names, location names, organization names and verbs, while [11] extracted person names and locations only.

Table 7. Confusion matrix of partial matches on the cross validation testing. Gold tags/columns and predicted tags/rows

| Gold\Predicted | TIM | PHO | DAT | LOC | PER | OUT |
|---|---|---|---|---|---|---|
| TIM | **549** | 1 | 4 | 1 | 3 | 37 |
| PHO | 0 | **220** | 0 | 0 | 0 | 8 |
| DAT | 17 | 0 | **921** | 1 | 0 | 74 |
| LOC | 0 | 1 | 1 | **850** | 12 | 216 |
| PER | 0 | 1 | 0 | 3 | **840** | 194 |
| OUT | 42 | 20 | 70 | 99 | 0 | **51073** |

Table 8. Strict matches on all labels (Left) and partial matches on the main labels (Right)

| Strict matches | | | | Partial matches | |
|---|---|---|---|---|---|
| Tag | Score | Tag | Score | Tag | Score |
| B-TIM | 86.95 | I-TIM | 84.78 | TIM | 91.27 |
| B-PHO | 93.14 | I-PHO | 90.32 | PHO | 93.62 |
| B-DAT | 92.89 | I-DAT | 69.43 | DAT | 91.69 |
| B-LOC | 81.47 | I-LOC | 87.94 | LOC | 83.58 |
| B-PER | 86.24 | I-PER | 68.83 | PER | 88.79 |
| O | 99.19 | **Total** | **86.44** | **Total** | **88.85** |

Table 9. Comparison with other SMS NER systems

| System | Language | Size | F-score |
|---|---|---|---|
| [7] | Chinese | 1000 | 61 |
| [11] | English | ? | 88 |
| This paper | Swedish | 4500 | 86 |

## 10. Conclusion and Future Work

We have presented a named entity recognition system based on the combination of regular expressions and corpus-driven classifiers. One of the major roadblocks we encountered to apply classical machine-

learning techniques lied in the collection a large SMS corpus. This proved very difficult due to the sensitive and personal nature of SMSes. The lack of SMS corpora limits the possibility of training a good model that is well rounded and can deal with most cases. In addition, there are inherent properties of SMSes that make the recognition complex:

- Our POS tagger is trained on newspaper text, whose style and language are quite different from those found in text messages. As a consequence, the tagger accuracy is significantly degraded.
- The brevity and lack of context in text messages impairs the ability to extract information at a high level.

Although we had also to cope with cellular telephone limitations on processing power, storage, and memory, we showed it was possible to reach accuracies competitive with those reported on other kinds of text. We also showed that it was possible to complement existing regex-based implementations with a machine-learning classifier and improve the overall system performance without severe penalties in terms of CPU or memory. This paves the way for a possible replacement of regular expressions with classifiers.

The brevity and lack of context is one of the major difficulties when applying NER on single SMS. We believe that there is a potential in exploring sequences of SMS (conversations) between users as a source of context.

Other contextual information sources specific to cellular telephones are telephone books or call logs for number or name references. Location information can be used both to localize *point of interest* (POI) search as well as for the disambiguation of generic POIs.

## References

[1] Bikel D. M., Miller S., Schwartz R., & Weischedel R. (1997). Nymble: High-performance learning name-finder. In *Proceedings of the Fifth ANLP Conference*, pages 194–201.

[2] Carlberger J., & Kann V. (1999). Implementing an efficient part- of-speech tagger. *Software – Practice and Experience*, 29(2):815–832.

[3] Chieu H. L., & Ng H. T. (2003). Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL-2003*, pages 160–163.

[4] Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., & Lin C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

[5] Florian R., Ittycheriah A., Jing H., & Zhang T. (2003). Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003*, pages 168–171.

[6] Hård af Segerstad Y. (2002). *Use and adaptation of written language to the conditions of computer-mediated communication*. Doctoral thesis, Göteborg University.

[7] Jiang H., Wang X., & Tian J. (2010). Second-order HMM for event extraction from short message. In *Proceedings of NLDB*, pages 149–156.

[8] Kudoh T., & Matsumoto Y. (2000). Use of support vector learning for chunk identification. In *Proceedings of CoNLL-2000 and LLL- 2000*, pages 142–144.

[9] Mikheev A., Moens M., & Grover C. (1999). Named entity recognition without gazetteers. In *Proceedings of EACL'99*.

[10] OpenNLP. (2004). A package of Java-based NLP tools. http: //opennlp.sourceforge.net/.

[11] Polifroni J., Kiss I., & Adler M. (2010). Bootstrapping named entity extraction for the creation of mobile services. In *Proceedings of LREC*.

[12] Tjong Kim Sang E. F. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158.

[13] Tjong Kim Sang E. F., & De Meulder F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

# Hyperlocal Event Extraction of Future Events

Tobias Arrskog, Peter Exner, Håkan Jonsson, Peter Norlander, and Pierre Nugues

Department of Computer Science, Lund University
Advanced Application Labs, Sony Mobile Communications
{tobias.arrskog,peter.norlander}@gmail.com
hakan.jonsson@sonymobile.com
{peter.exner,pierre.nugues}@cs.lth.se

**Abstract.** From metropolitan areas to tiny villages, there is a wide variety of organizers of cultural, business, entertainment, and social events. These organizers publish such information to an equally wide variety of sources. Every source of published events uses its own document structure and provides different sets of information. This raises significant customization issues. This paper explores the possibilities of extracting future events from a wide range of web sources, to determine if the document structure and content can be exploited for time-efficient hyperlocal event scraping. We report on two experimental knowledge-driven, pattern-based programs that scrape events from web pages using both their content and structure.

## 1 Introduction

There has been considerable work on extracting events from text available from the web; see [1] for a collection of recent works. A variety of techniques have been reported: [2] used successfully data-driven approaches for the extraction of news events while knowledge-driven approaches have been applied to extract biomedical [3], historical [4], or financial events [5] among others.

Much previous research focuses on using the body text of the document, while some authors also use the document structure. For example, [4] apply semantic role labelling to unstructured Wikipedia text while [6] use both the document structure and body text to extract events from the same source.

The focus of this paper is on extracting future events using the body text of web pages as well as their DOM structure when the content has multiple levels of structure. We naturally use the body text from the web page as itthat contains essential information, e.g. time, date, and location instances. We also exploit the DOM structure as a source of information. Although HTML embeds some sort of structure, the actual structure is not homogeneous across websites. We report on the problem of extracting event information from a variety of web pages and we describe two systems we implemented and the results we obtained. .

### 1.1 Properties of Local Events

The events we are interested in are those that typically appear in calendars and listings, such as cultural, entertainment, educational, social, business (exhibitions, conferences), and sport events, that attract athe general and large public may have an interest in.

The end goal of this project is to be able to serve users with information about events that match their current interest and context, e.g. using location-based search, by aggregating these events from hyperlocal sources.

Event aggregators already exist, e.g. *Eventful* and *Upcoming*, that collect and publish event information, but they tend to only gather information about major events in cooperation with organizers or publishers. By contrast, we want to extract existing information directly from the publisher.

The main challenge is time-efficient scaling since there is a great number of hyperlocal organizers and sources as well as variations in the formats and DOM structure of the sources and ambiguity. We may also have to deal with missing, ambiguous, or contradictory information. For example, locations can appear in the title:

> Concert – Bruce Springsteen (This time in the new arena),

and contradict the location indicated elsewhere. Another example is a title:

> Outdoor dining now every Friday and Saturday

containing date information which narrows or sometimes contradicts the dates indicated elsewhere on the page.

FThe domain we are interested deals with future events form. This is a very wide area, where only few historically-annotated data is available. This makes a statistical approach problematic, at least initially. Instead, we chose a knowledge-driven, pattern-based approach, where we process both the structure of HTML documents and their content. We analyzse the content using knowledge of the event domain, e.g. event keywords.

In this paper, we report on the problem of extracting event information from given web pages and we describe two systems we implemented and the results we obtained.

### 1.2 Applications and Requirements for Event Structures

From the possible properties of an event, wWe chose to extract the *title, date, time, location, event reference (source)* and *publisher* which answers the *wWhen, where*, and *what* questions aboutof thean event. These are however the most basic attributes, and for a useful application, further information could be extracted, including topic, organizer, cost and target audience.

We set aside In this paper, we do not cover the semantic representation of event data, but future research may need to address representing the above attributes in existing event data models.

## 2 System Architecture

### 2.1 Designing a Simple Scraper

For each site in the list, we created a unique script. These scripts contained a hand-crafted set of rules to extract the correct information for that specific site. This may require a good deal of manual effort as we naturally have toTo expand the list of additional hand-crafted scripts is required, which leads to high costs when scaling to multiplemany sources..

In order to limit scaling costs, the scripts need to be simplistic. For this reason, we decided to A chosen limit ation was that the internal structure of the information in the events needs to be the same between each other, so that a small set of rules can extract the information from all the events.

### 2.2 Designing a Generic Scraper

We investigated if it would be possible to create a generic scraper which could handle all websites without manual labour.

The first step to generically scrape a website is to find all the pages that contain events. This is currently done using domain knowledge, i.e. the system is given only pages which are known to contain events. The possibilities to find pages without manual labour is further discussed in Sect. 5. The system uses six steps to scrape the events from a given web page. Figure 1 shows the system architecture. We implemented the first three steps using the ad-hoc scripts of Sect. 2.1.

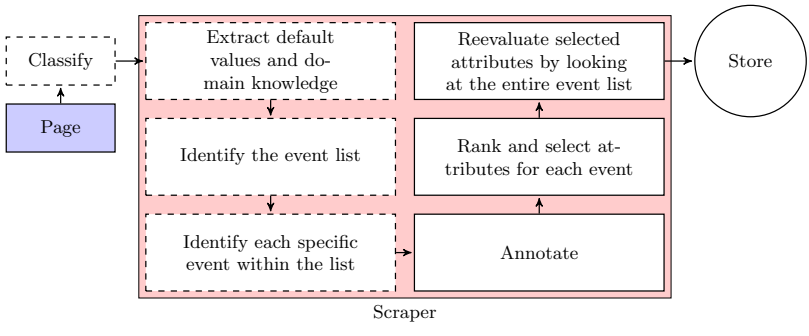



**Fig. 1.** The implemented generic scraper. Dashed boxes use manually written, site-dependent scripts.

### 2.3 Attribute Annotation and Interpretation

The system uses rules to annotate and interpret text. The benefit of a rule-based system is that it can both parse the text and create structured data. As previous work suggests, extracting the time and date of events can be solved through rules. While problematic, the system is able to extract named entities, for example named locations as well. To do this, the system uses three major rules:

1. Keyword detection preceding a named location, e.g looking for *location:* or *arena:*
2. Keyword detection succeeding a named location, for example a city
3. Structured keyword detection preceding a named location. e.g. look for *location* or *arena* when isolated in a separate structure. As an example: ***location*** *Boston* which corresponds to "`<b>location</b> Boston`" using HTML tags.

When the rules above return a named location, we query it against a named location database. Using these rules and a database lookup, we can minimize the false positives.

### 2.4 Attribute Ranking and Selection

The system uses domain knowledge to choose what data to extract:

- The system extracts only *one* title and chooses the most visually distinguished text it can, implied by the DOM structure
- Dates and times are following a hierarchy of complexity, where it takes those of highest complexity first. Some sites used a structure where event structures were grouped by date. To avoid false positives with dates in these event structures, the scraper choose dates between the event structures if less than half of the event structures contained dates.
- The extraction of the location for the event was done in the following order: If the event structure contained a location coordinate, choose it. Otherwise use a default location. If the event site had no default location, use the most commonly referred city in the event structure.

## 3 Evaluation

### 3.1 Scoring

We evaluated the performances of the simple and generic scrapers and we compared them with a scoring defined in Table 1.

**Table 1.** Criteria for full and partial scoring for the test set.

| | Full match |
|---|---|
| Title | Lexicographic distance to correct = 0 |
| Date | Resulting date(s) equal to correct date(s) |
| Time | Resulting start time equals correct start time (minute) |
| Location | Result within 1000 m of correct |

| | Partial match |
|---|---|
| Title | Result contains correct title |
| Date | Full match or if result contains at least one of correct date(s) |
| Time | Full match or if result contains at least one of correct start time(s) |
| Location | Result within 5000 m of correct |

### 3.2 Training

At the start of the project, we gathered a training set composed of nine different event sites found in the Lund and Malmö area, Sweden. With the help of the training set, we could change the rules or add new ones and easily monitor their overall effect. This concerned both the rules of the annotator, scraper, and the location lookup.

### 3.3 Evaluation

In order to evaluate the system, we gathered a test set of nine, previously unseen, event web sites. The goal was to extract information about all (max. 30) events. The tests were conducted in three parts.

1. In the first part, we used the generic scraper (Sect. 2.2);
2. In the second one, we built simple scrapers (Sect. 2.1) for each of the test sites.
3. We extracted the events manually by hand in the third part.

The results from the first two parts were then compared against the third.

The generic scraper and the simple scrapers were compared in how accurately they extracted the title, date, time, and location of the event. The time of the setup was also compared for both the generic and simple scrapers.

We built a simple scraper for each site specifically to extract the text containing the title, date, time, and the location. The text strings containing the dates and times were then sent to the same algorithm that the generic scraper uses to parse the date and time. Once the text containing the location is extracted, we use the same location lookup in all the scrapers.

### 3.4 Bias Between the Training and Test Sets

The sites in the training set were all composed of a list with events where all the necessary information (title, date, time, location) could be found. In the

**Table 2.** $F_1$ score for full and partial match on test data for the generic scraper.

| | Full | | | | | Partial | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Site | Title | Date | Time | Location | Average | Title | Date | Time | Location | Average |
| lu | 0.0 | 0.967 | 0.767 | 0.433 | 0.542 | 0.4 | 0.967 | 0.933 | 0.633 | 0.733 |
| mah | 0.068 | 1.0 | 0.0 | 0.6 | 0.417 | 0.915 | 1.0 | 1.0 | 1.0 | 0.979 |
| babel | 0.0 | 0.818 | 0.0 | 1.0 | 0.830 | 1.0 | 0.909 | 0.818 | 1.0 | 0.932 |
| lund.cc | 1.0 | 0.667 | 1.0 | 0.652 | 0.714 | 1.0 | 0.967 | 1.0 | 0.652 | 0.905 |
| möllan | 0.0 | 0.857 | 1.0 | 1.0 | 0.75 | 0.0 | 0.857 | 1.0 | 1.0 | 0.714 |
| nsf | 1.0 | 1.0 | 1.0 | 0.0 | 0.673 | 1.0 | 1.0 | 1.0 | 0.286 | 0.822 |
| malmö.com | 1.0 | 1.0 | 0 | 0.691 | 0.543 | 1.0 | 1.0 | 0 | 0.963 | 0.741 |
| burlöv | 0.889 | 0.75 | 0.333 | 0.2 | 0.369 | 1.0 | 0.875 | 0.333 | 0.2 | 0.602 |
| dsek | 0.0 | 0.2 | 0.444 | 0.833 | 0.588 | 1.0 | 0.2 | 1.0 | 0.833 | 0.758 |
| Average $F_1$ | 0.440 | 0.807 | 0.505 | 0.601 | 0.603 | 0.813 | 0.864 | 0.787 | 0.730 | 0.799 |

**Table 3.** $F_1$ score for full match on test data for the generic scraper without loading the event details page.

| | Full | | | | Partial | | | |
|---|---|---|---|---|---|---|---|---|
| Site | Title | Date | Time | Location | Title | Date | Time | Location |
| lu | 1.0 | 1.0 | 0.967 | N/A | 1.0 | 1.0 | 0.967 | N/A |
| mah | 0.967 | 0.929 | 1.0 | N/A | 0.967 | 0.929 | 1.0 | N/A |
| babel | 0.0 | 0.0 | N/A | 1.0 | 1.0 | 0.0 | N/A | 1.0 |

**Table 4.** $F_1$ score for full and partial match on test data for the simple scraper.

| | Full | | | | | Partial | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Site | Title | Date | Time | Location | Average | Title | Date | Time | Location | Average |
| lu | 1.0 | 0.967 | 0.967 | 0.267 | 0.800 | 1.0 | 1.0 | 1.0 | 0.667 | 0.917 |
| mah | 1.0 | 1.0 | 0.0 | 0.7 | 0.675 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| babel | 0.0 | 0.7 | 0.211 | 1.0 | 0.478 | 1.0 | 0.7 | 0.632 | 1.0 | 0.833 |
| lund.cc | 1.0 | 0.667 | 1.0 | 0.622 | 0.822 | 1.0 | 0.967 | 1.0 | 0.622 | 0.897 |
| möllan | 0.857 | 0.667 | 1.0 | 1.0 | 0.881 | 1.0 | 0.833 | 1.0 | 1.0 | 0.959 |
| nsf | 1.0 | 1.0 | 1.0 | 0.0 | 0.75 | 1.0 | 1.0 | 1.0 | 0.0 | 0.75 |
| malmö.com | 1.0 | 1.0 | 0.0 | 0.823 | 0.706 | 1.0 | 1.0 | 0 | 0.912 | 0.728 |
| burlöv | 1.0 | 1.0 | 0.0 | 0.0 | 0.5 | 1.0 | 1.0 | 0.0 | 0.0 | 0.5 |
| dsek | 0.952 | 0.706 | 0.778 | 1.0 | 0.859 | 0.952 | 0.706 | 0.889 | 1.0 | 0.887 |
| Average $F_1$ | 0.868 | 0.856 | 0.551 | 0.601 | 0.719 | 0.995 | 0.912 | 0.725 | 0.689 | 0.83 |

**Table 5.** Time taken for the setup for the test sites.

| Site | Generic | Simple | Manual |
|---|---|---|---|
| lu | 23 min | 83 min | 60 min |
| mah | 7 min | 24 min | 68 min |
| babel | 11 min | 59 min | 15 min |
| lund.cc | 9 min | 13 min | 60 min |
| möllan | 2 min | 31 min | 13 min |
| nsf | 5 min | 24 min | 15 min |
| malmö.com | 31 min | 63 min | 35 min |
| burlöv | 10 min | 30 min | 22 min |
| dsek | 11 min | 23 min | 21 min |
| Average | 12 min | 39 min | 34 min |

test set, most of the sites had a structure that did not have all the required information: Each event had a separate page with all the information, the event details page. The information on the event details page was not composed of the typical compact structured form but rather had more body text. Of the nine sites in the test set, three sites (lund.cc, nsf, dsek) did not require an event details page for the necessary information. But the information on the sites nsf and dsek were in their structure more comparable to a body text. A concept to handle this is presented in Sect. 4.1 that concerns the extraction of the title.

## 4 Conclusion

The setup for the generic scraper took on average 12 minutes, compared to creating a simple scraper for each site that took on average 39 minutes (Table 5). The setup for the generic scraper is more than three times faster than creating a simple scraper for each site. This can be compared to the pure manual labor which took on average 34 minutes per site, thus both scrapers essentially have a pay back time of one pass.

### 4.1 Title

The generic scraper performs rather poorly on the test set while it shows better results on the training set. This is either due to a training overfit or a significant mismatch between the training and test sites. Sect. 3.4 analyzes the mistakes and discusses this problem. When using the system on these pages without loading, they do yield better results, as shown in Table 3. The rest of the failing test sites failed because the system looked to much in the structure where it should have analyzed the layout instead, i.e. it chose links when it should have chosen the ones which were more visually prominent.

### 4.2 Date

The simple scraper is 5% better on the date identification than the generic scraper on average for both the full and partial matches. Examining the scores

for the full match more closely, (Tables 2 and 4), the score for the generic is the same or better than the score for the simple scraper for every site except burlöv and dsek. We even observe a complete failure for dsek. We investigated it and we discovered that dsek expressed the dates relative to the current date e.g. *today*, *tomorrow*. This wasn't implemented yet which made the generic scraper pick another strategy for picking dates, as a result the correct dates were forfeited.

### 4.3 Time

The average scores for the time extraction between the generic and the simple scrapers are rather similar. The system does finds the correct times but does report many false positives, which according to the scoring set in Sect. 3.1 yields only a partial match. The system tends to over detect times. We programmed it to prefer times coupled with dates over solitary times but in the test set, it seems it was rather common to have time and dates further apart. This makes the system choose all times, where it should have chosen a subset. Another pattern was also found: for some sites, the system returned both start and end time separately which shows that the system is lacking rules to bind start and end times together.

### 4.4 Location

The difference between simple and generic scraper is negligible and the problem of location is less about selection and more about actually find and understand the named locations (Tables 2 and 4). The system uses assumed knowledge to fill in what is left out of the events, i.e. knows city, region or location which it can use to fallback to or base the search around. Using this assumed knowledge has proved useful when looking at babel, möllan, dsek, lu and mah and this should hold true on all hyperlocal websites. Even if the system has some basic knowledge about the web page, the location annotation and selection still has problems with disambiguation. This disambiguation problem is partly rooted in the fact that the named locations are within the domain knowledge of the site. As an example, a university website might write lecture halls or class rooms as the location of the event. These named locations could have the same name as pub in another city, a scientist or simply nonexistent in any named location database.

### 4.5 Final Words

At the end of the test cycle, however, we considered that an generic scraper is not only possible to do, but in some cases even better than a simple one. The hardest problem with scraping sites is not necessarily to understand the structure, even if vague. The problem for a scraper is rather to understand what can only be described as domain knowledge. Sites uses a lot of assumed knowledge which can be hard to understand for a machine or even if its understanding could be

completely wrong in the context. For example, lecture halls can be named the same as a pub in the same region, making it hard for a system to determine if the location is correct or not. This might be attainable with better heuristics, e.g. if the location lookup can be made with some hierarchical solution and domain knowledge can be extracted from the sites prior to the extraction of events.

## 5 Future Work

### 5.1 Page Classification

On the Internet, sites show a significant variation and most of them do not contain entertainment events. Therefore a first step in a generic system, the dashed box "Classify" in Figure 1, would be to identify *if* the input web page contains events. If it does not, it makes no sense to scrape it and doing so could even lead to false positives. If web pages could be classified with reasonable certainty, it could also be used with a crawler to create an endless supply of event pages to scrape.

### 5.2 Exploring Repetitiveness

To solve the dashed box "Identify the event list" shown in Figure 1, we investigated the repetitiveness of the event list. With the help of weighing in structural elements, e.g. P, STRONG, H3, it yielded some interesting results on small sites. This technique can potentially be further refined by calibrating weights if the page is annotated using what is described in Sect. 2.3.

### 5.3 Rank and Select with Help of Layout

While the system uses a very limited rank and selection based on an implied layout for title (prefer H3, H2 etc. over raw text), it would be interesting to have the selection fully use layouts. To attract attention and to create desire, the vital information about an event are among the first things the reader is supposed to notice and comprehend. Thus it is usually presented in a visually distinguishing way. This can be achieved by coloring the text differently, making it larger, or simply in a different font or typing. This layout is bundled within the HTML document, possibly modified by the CSS, thus looking at these clues with some heuristics allows to find the visually distinguishing sentences [7]. As an example, an event might use a H3 element for the title, bold for the location, or it might have another background color for the date. If the entire system would use layout to aid the selection we believe that the system will perform better and will yield less false positives.

## References

1. Hogenboom, F., Frasincar, F., Kaymak, U., de Jong, F.: An Overview of Event Extraction from Text. In van Erp, M., van Hage, W.R., Hollink, L., Jameson, A., Troncy, R., eds.: Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011). Volume 779 of CEUR Workshop Proceedings., CEUR-WS.org (2011) 48–57
2. Liu, M., Liu, Y., Xiang, L., Chen, X., Yang, Q.: Extracting key entities and significant events from online daily news. In Fyfe, C., Kim, D., Lee, S.Y., Yin, H., eds.: Intelligent Data Engineering and Automated Learning - IDEAL 2008. Volume 5326 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2008) 201–209
3. Chun, H.w., Hwang, Y.s., Rim, H.C.: Unsupervised event extraction from biomedical literature using co-occurrence information and basic patterns. In: Proceedings of the First international joint conference on Natural Language Processing. IJCNLP'04, Berlin, Heidelberg, Springer-Verlag (2005) 777–786
4. Exner, P., Nugues, P.: Using Semantic Role Labeling to Extract Events from Wikipedia. In van Erp, M., van Hage, W.R., Hollink, L., Jameson, A., Troncy, R., eds.: Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011). Volume 779 of CEUR Workshop Proceedings., CEUR-WS.org (2011) 38–47
5. Borsje, J., Hogenboom, F., Frasincar, F.: Semi-automatic financial events discovery based on lexico-semantic patterns. Int. J. Web Eng. Technol. **6**(2) (January 2010) 115–140
6. Hienert, D., Luciano, F.: Extraction of historical events from wikipedia. In: Proceedings of the First International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, CEUR-WS.org (2012)
7. Cai, D., Yu, S., Wen, J.R., Ma, W.Y.: Extracting content structure for web pages based on visual representation. In: Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications. APWeb'03, Berlin, Heidelberg, Springer-Verlag (2003) 406–417

# Proximates – A Social Context Engine

Håkan Jonsson and Pierre Nugues

Lund University, LTH, Box 118, SE-221 00, Lund, Sweden
{hakan,pierre.nugues}@cs.lth.se

**Abstract.** Several studies have shown the value of using proximity data to understand the social context of users. To simplify the use of social context in application development we have developed Proximates, a social context engine for mobile phones. It scans nearby Bluetooth peers to determine what devices are in proximity. We map Bluetooth MAC ids to user identities on existing social networks which then allows Proximates to infer the social context of the user. The main contribution of Proximates is its use of link attributes retrieved from Facebook for granular relationship classification. We also show that Proximates can bridge the gap between physical and digital social interactions, by showing that it can be used to measure how much time a user spends in physical proximity with his Facebook friends. In this paper we present the architecture and initial experimental results on deployment usability aspects of users of an example application. We also discuss using location for proximity detection versus direct sensing using Bluetooth.

**Keywords:** Mobile Phone Sensing, Proximity, Social Context, Social Sensing.

## 1 Introduction

The purpose of middleware for social context is to simplify development of applications that use social context. By social context we mean individuals and groups in proximity of a user and the relation of the user to the individual and group, for example family, co-workers, friends, sometimes referred to as pervasive social context [17]. Modeling a user's social context is not trivial. It requires knowledge about privacy, mobile sensing, power efficient data collection, data cleaning and analysis, clustering, etc. A mobile software component that addresses all this complexity is vital to save development effort and cost. The developer will then be able to focus on the task of using social context rather than extracting it.

There has been several studies investigating the relations between online social networks such as Facebook and social networks spanned by physical proximity or co-location retrieved from mobile phones. An early major research project into using proximity data for understanding a user's social context was the Reality Mining project [9]. This project studied social changes in organizations adopting proximity based applications, but also suggested consumer oriented applications, e.g. Social Serendipity [8], but did not include integration with an online social network. The SocioPatterns [1] project combined proximity sensing using directional RFID with online social networks, including Facebook. Later [16], this is used for link prediction in the proximity

M.J. O'Grady et al. (Eds.): AmI 2013 Workshops, CCIS 413, pp. 230–239, 2013.

network. Cranshaw et al. [7] model the social context of locations a user visits to do link prediction in the Facebook network. To this end they use location trails collected from GPS and Wifi networks on mobile phones. However, as shown by several studies [2, 6, 14] spatio-temporal granularity makes all the difference in modelling human social interactions, and location sensing rather than proximity sensing is not granular enough for our purposes. The Lausanne Data Collection Campaign has given rise to several important studies in this area, such as [5, 10, 13] but does not include Facebook data. WhozThat [3] use both sensed proximity and social network ids to bridge the gap between physical and online social network identities. However, the simplicity of the protocol raised some serious privacy issues as noted by the author, and it was not deployed in field trials with smartphone users. SocialFusion [4] address the privacy issues of WhozThat by proposing alternatives to K-anonymity for anonymization.

Middleware to address the complexities of developing pervasive social networking applications has been the topic of several studies as well. In a survey of mobile social network middlewares [12], requirements on such middleware is defined, which we use in the description of Proximate's architecture. Mokhtar et al. [15] suggest using Bluetooth for proximity detection. They discuss different potential deployment strategies of their architecture, and results are based on simulations using Reality Mining data and a social network derived from text message interactions.

In this paper we build upon the concept developed in Serendipity, to use the Bluetooth radio transmitter as a carrier of identity. In Serendipity a separate digital social network was created. In our project Proximates, we use Facebook and phone number as the digital identity of users and bridge it to the physical identities emitted by the users' devices. This allows us to analyze the relation between Facebook friendship and physical proximity, e.g. how much time a user spends with Facebook friends. We believe that Proximates can be used across many applications and used to build a corpus of social context data that can be shared across researchers. To satisfy users need of privacy as stated in the Obvious Data Usage Principle, we need to build value in proximity data. Proximates does this by supporting the bridging of physical and digital identities with low latency. We will use Proximates to study users perceptions of privacy regarding this bridging, architectures that satisfy scaling of research applications to large numbers of users, and spatio-temporal aspects of social dynamics.

In the first section(System Architecture) we present the architecture of Proximates (Figure 1) and how it bridges the gap between physical proximity space and online social networks. In section Applications and Results we present some early experimental results from a user study and example applications that was built on Proximates for the study.

## 2 The Proximates Social Context Engine

The purpose of Proximates is to simplify development of mobile phone applications that use social context. By social context we mean individuals and groups in proximity of a user and the relation of the user to the individual and group, for example family, co-workers, friends. By social context classification we mean the inference of the relationship class of such an individual or group. Which user identities and social networks

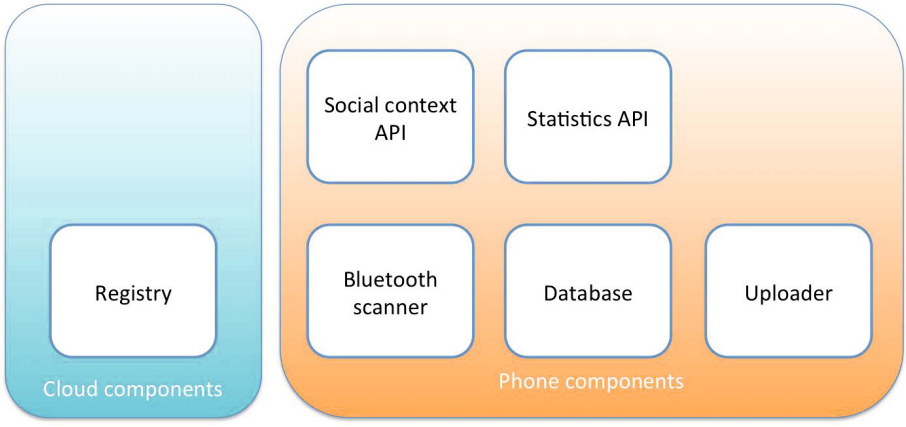



**Fig. 1.** Architecture of Proximates

to use is application specific, but can be shared across applications if desired. In its current deployment, Proximates use Facebook ids and phone numbers to identify users across applications.

### 2.1 System Architecture

Proximates consists of six components: a Bluetooth scanner, a database, two APIs, an uploader and a device registry.

**Bluetooth Scanner and Social Context API Components.** The Bluetooth scanner is a service that runs in the background on a mobile device. Periodically it performs a Bluetooth scan for nearby Bluetooth peers with the phone device class, and stores the result in the database. The data includes MAC ids, signal strength, and device class.

The social context API is a background service that carries out mining on the stored data in the database and triggers on events from the Bluetooth scanner. It performs smoothing of Bluetooth scans over time and group them for easy access and Bluetooth MAC ids are mapped to user ids. An API to application developers that allows applications to get notifications when a contact or group of contacts is in proximity, or when a user's social context change. The social context of a user is a ranked list of relationship labels, where the top label is the most common relation of the peers in proximity to the user, over a scan period. For example, if there are five peers in a scan where three are known to the user and two of them are classified as Family and three of them are classified as Colleagues according to the user's Facebook friend list, then the top ranked relation will be Colleagues, the second Family and the third Unknown.

To know the relation between a user and its friends, Facebook friend lists are used. When an application is notified of the proximity of a person, it can retrieve the classification of the relation to that person.

We use the labelled data to train classifiers of social context for users who don't use friend lists. The training of classifiers is ongoing work and results will be presented in future papers.

**Registry Component.** The purpose of the registry is to map Bluetooth MAC ids to any public ids of its owner. The public owner ids can be any application specific ids or general public ids, such as phone numbers or Facebook ids. It is up to the application that registers the device and its owner to determine which user ids to register and whether they should be hashed or not. Hashing makes it hard to make a lookup from a Bluetooth MAC id to a useful user id unless the user id is already known.

The registry is a web service and exposes a simple JSON REST API. The registry API performs all the operations on a single resource: the device. There are methods for adding, update, deleting devices, as well as retrieving a single device or a list of devices providing user ids and MAC ids as query parameters.

Applications are encouraged to cache results from device queries in order to minimize data traffic, server load, and power consumption. For known contacts, for example phonebook contacts, the Bluetooth MAC ids can be cached for a long time since they are not likely to change often. Some applications will not know the id they are looking for in advance and will need to lookup any new peers that are in proximity. These results should also be cached since transient peers often appear in at least some scans.

**Database, Statistics API and Uploader.** The database stores collected sensor data and events. The statistics API allows the application developer to query historical information, for example retrieve the most frequently occurring people, groups of people or social contexts of the user, over a specific time frame. The uploader pushes the stored data to a server. The uploader is an optional component that is deployed if Proximates is used for research applications, for example in computational social science, where extensive data logging is needed for analysis.

### 2.2 Requirements

As a middleware intended for real world deployment, Proximates needs to fulfill several requirements that are common to middleware for mobile social networking and pervasive social context. In the survey of mobile social middleware [12], the aspects below are analyzed, and we use them here as reference requirements. We also use the two social context modeling requirements [18] defined by Tran et al.

**Simplification of Development Process.** Modelling social context requires knowledge about privacy, mobile sensing, power efficient data collection, data cleaning and analysis, stream processing, clustering, etc.The social context API for detecting proximity of people, groups and social context is a very simple and high-level API.

**Energy Efficiency.** Power consumption is a major concern for opportunistic sensing applications. Recent availability of dedicated sensor processing subsystems in smartphone chipsets is improving the situation by allowing continuous sensing with low

power consumption. However, application sensing of network data from Bluetooth, 3G and Wifi still needs to be done in the application CPU.

Proximates uses Bluetooth to detect proximity. By using Bluetooth, proximity is sensed directly rather than indirectly through a translation to location coordinates and distance calculation. Bluetooth consumes less battery than GPS and Wifi, and only needs to be sampled periodically. Using Bluetooth rather than GPS or Wifi avoids both translation to location coordinates in the case of Wifi, and more importantly removes the need for frequent uploading of location data for all users for which we want to detect proximity.

The power consumption of scanning can be traded off with latency. If an application needs to be notified of a nearby person with low latency, power consumption increases. The latency in Proximates is configurable through setting of the scan rate. The default rate of 2 minutes makes the power consumption very low and in general not noticeable to the end user.

**Privacy.** By using hashed identifiers for people, for example Facebook IDs and phone numbers, Proximate applications require their users to already know the identities of the people they want to detect proximity of. This means they already need to be friends on Facebook or to already have their phone numbers. This is a more secure approach than the one used in WhozThat which transmits Facebook IDs in clear text. It is up to the application to determine whether to used hashed IDs or not, depending on requirements. For applications that do not need the access to the registry to be open, for example in the case where the ID is entirely application specific, strict access control to the registry component can be enforced rather than providing shared access across applications.

However, the attitudes of users regarding mapping their phones' Bluetooth MAC ids to personal identities, such as Facebook identities, is unknown. As far as we know, no such studies has been done. Several studies on privacy aspects of location sharing has been done, but we cannot assume they apply directly to proximity. We believe that sharing of your social network identity connected to your Bluetooth MAC id is less sensitive than sharing location data, since within Bluetooth range it is hard to hide your identity anyhow. At least in the non-public case, where only people who know you can detect you when in proximity. This remains to be verified by user studies.

**Scalability and Distributed Architecture.** The architecture of Proximates is very simple compared to most mobile social networking platforms since it focuses on a specific problem, uses proximity sensing, and delegates management of social networks to the original social network services rather than aggregates. Using direct proximity sensing rather than via location makes it possible to do the sensing directly on the device. This eliminates scalability problems associated with pairwise distance computations. The registry is a centralized component, but it is not subject to heavy loads since registrations seldom change which allows for long caching in clients.

**Heterogeneity and Dynamicity of Mobile Environments.** Performing the proximity detection directly on the device removes the need for continuous network coverage and data connection, making Proximates insensitive to networks signal strength fluctuations. Regarding heterogenous evironment, this is ignored by selecting Android 4.0.4 or later as the target environment.

**Social Context Modelling.** Tran et al. defined requirements [18] for social context modeling. We show how Proximates fulfill these and elaborate on them:

> Social context needs to explicitly capture constructed relationships and interaction constraints between actors. This set of constructed relationships and constraints needs to be managed, and modeled subjectively from an actor's perspective. The architecture of context-aware systems needs to externalize the management of social context from the implementation of actors.

Proximates explicitly models relationships through the integrated social networks, defined by the application. Currently integrated networks are Facebook and phone contacts. The user manages his Facebook relations using the Facebook service and his phone contacts through the phone book application, and are thus externalized. Interaction constraints are not specifically captured, but are left to the application since these are application specific. The relationships are modeled subjectively since Facebook friend lists is managed by the user and only visible to him.

> The architecture also needs to support the adaptability of social context, and needs to be easily deployable.

The complexity and cost of integrating and deploying a social context engine in commercial applications must be low. Many companies are yet to understand the potential benefits of context aware applications. This means that a small and simple component that solves a specific problem in existing infrastructure is preferable to a complex system that solves a wide range of problems. Additional context information should be added through integration of additional simple components that integrate well. It should also utilize existing infrastructure and services, e.g. Facebook for management of a user's social graph. Furthermore, the social context model must be simple and usable across several applications. Proximates fulfill these requirements through its simplicity and integration with existing services.

**Additional Requirements.** In addition to the requirements used in [12] we have further requirements on Proximates.

- Low latency for proximity detection is a requirement for some applications, for example reminder applications triggered by proximity to a person. This requirement makes it impossible to use location for proximity detection.
- Robustness, i.e. not needing to rely on GPS satellite visibility, availability of nearby wifi access points and a network data connection, also makes us choose direct proximity sensing.
- Finally, accuracy of location based methods is at best 10 meters indoors which is not enough to detect actual social interactions. Using Bluetooth, 10 meters is the maximum distance. Also, as shown by Cattuto et al. [2, 6] spatio-temporal granularity makes all the difference in modeling human social interactions, and location sensing rather than proximity sensing is not granular enough for our purposes.

# 3 Applications and Results

## 3.1 SmartTodos

Proximates was used to develop SmartTodos, a contextual reminder application, that for allows a user to add a reminder that will trigger when he is in close proximity of a specific contact or Facebook friend, in addition to reminders based on time and location triggers. It was distributed to about a 100 users who installed it and who could also send invites to others. We sampled seven users out of this population to make a usability study. The users in the sample were advanced smartphone users with university degrees.

Task completion time and requests for help was measured for seven different tasks as shown in figure 2. Task 1 is the task of setting up Proximates, while the other tasks are related to SmartTodos as such, for example the creation of location alarms. It is clear from the study that setting up Proximates is a hard task. It includes the following manual steps for the users: Signing into Facebook and accepting requested permissions, entering phone number, accepting enabling of Bluetooth and location services if not enabled, and setting Bluetooth visibility timeout to infinity. The most complicated step is the last one. This step should not be needed at all, but exists due to a bug in Android. Entering of phone number is often needed since most operators do not provide the information on SIM cards. Preloading of Proximates, an option only available to phone OEMs, can remove some of these obstacles, but it is clear that it is important to reduce the complexity of these tasks in Proximates.

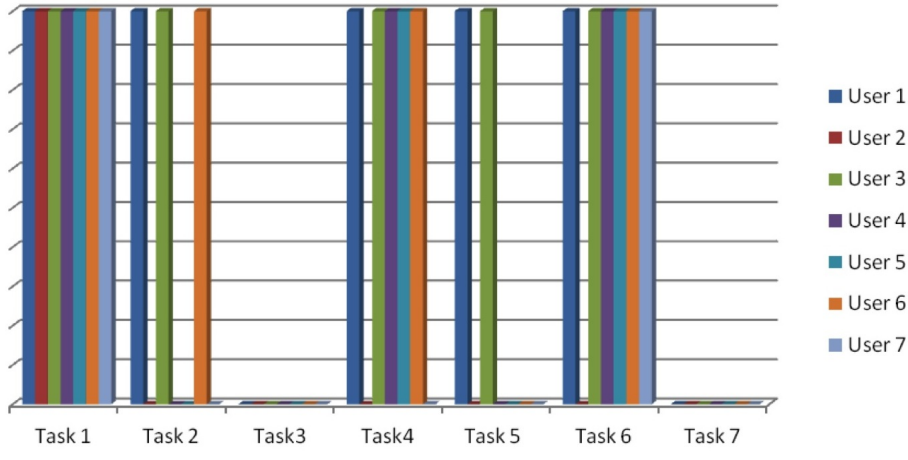



**Fig. 2.** Requests for help

## 3.2 Data Collection

In the SmartTodos trial we deployed the application with the upload component. Proximates collected proximity and Facebook data and uploaded them to a server where they were stored for analysis. Bluetooth scans were performed every minute, and only

Bluetooth with device class "phone" were stored for analysis. In the period July 2012 to March 2013, 2,466,036 Bluetooth scans containing 84,793 peers were collected by 135 devices. 161 users were registered with Facebook id in the registry, and they had 29,979 friends in total. 99 users had no user defined friend lists while the median number of friend lists was 9 among the 62 others. In addition to the traditional informed consent trough terms of service agreement, The Obvious Data Usage Principle [11] was applied in the application design to make it clear to users what information was being collected and how it was used. All data collected is anonymized through hashing.

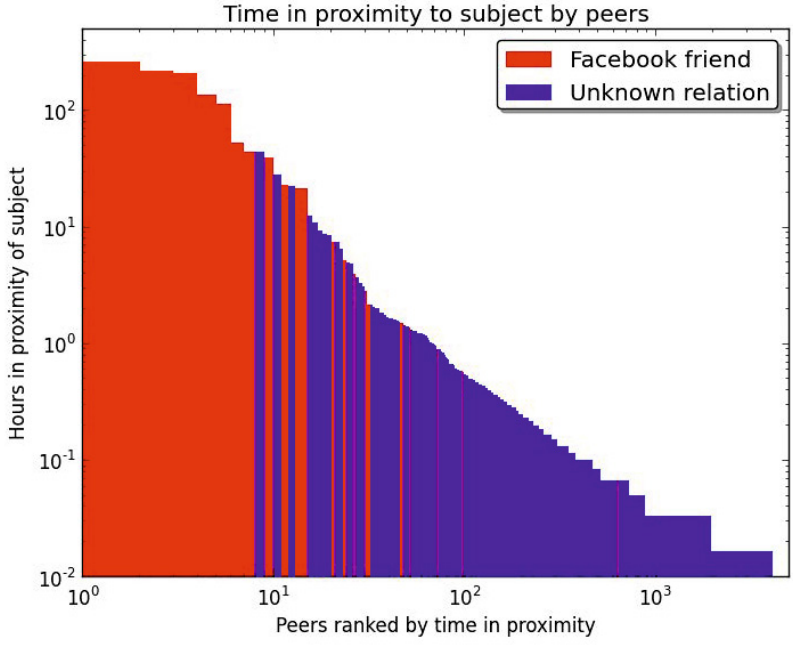



**Fig. 3.** Time in proximity of a single subject, and Facebook relationship

Figure 3 is a log-log plot of the time spent in proximity to one specific user versus the peers ranked by the same measure. In the plot we have colored each peer according to its relationship to the user. Red indicates a Facebook friendship relation, while blue means that the relationship is unknown, since that peer is not registered in the whoowns registry. A peer colored in blue could still be a Facebook friend of the user, but we have no information about this. This means that the time this user spent with Facebook friends was significantly larger than with friends with unknown relationships. An interpretation of this is that Facebook friendship is not only used to keep in touch with distant friends, but also a relationship users have with the people they actually spend time with. More work is needed to determine the generality of this result to other users.

## 4 Conclusion and Future Work

We have presented an architecture for proximity based services on smartphones, that is power efficient, easy to deploy, delegates social network management and scales well due to using direct proximity sensing rather than location and distance calculations. It also has other privacy properties than location sensing that needs further investigation.

We have shown that Proximates can be used in real world applications by means of the SmartTodos application.

Furthermore, we have shown that Proximates can be used for research applications by showing that we can measure the time users spend in proximity of Facebook friends. We also reported on how many users actually use Facebook friends lists. Further work is needed to investigate if Facebook friends list labels has the potential to be used as ground truth for classification of social context when Facebook friends labels are not available. We will also study how to improve usability by reducing the complexity of the setup of Bluetooth visibility and user ids in different social networks. We are about to launch SmartTodos on Google Play in order to scale up the amount of users and data collected.

**Acknowledgements.** This work was partly funded by the Industrial Excellence Center EASE – Embedded Applications Software Engineering, (http://ease.cs.lth.se) and the European 7th Framework Program, under grant VENTURI (FP7-288238).

## References

1. Alani, H., Szomszor, M., Cattuto, C., Van den Broeck, W., Correndo, G., Barrat, A.: Live social semantics. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 698–714. Springer, Heidelberg (2009), http://link.springer.com/chapter/10.1007/978-3-642-04930-9_44
2. Barrat, A., Cattuto, C.: Temporal networks of face-to-face human interactions. Temporal Networks, pp. 50–55 (2013), http://link.springer.com/chapter/10.1007/978-3-642-36461-7_10
3. Beach, A., Gartrell, M., Akkala, S., Elston, J., Kelley, J., Nishimoto, K., Ray, B., Razgulin, S., Sundaresan, K., Surendar, B., Terada, M., Han, R.: WhozThat? Evolving an ecosystem for context-aware mobile social networks. Network IEEE 22(4), 50–55 (2008), http://dx.doi.org/10.1109/MNET.2008.4579771
4. Beach, A., Gartrell, M., Xing, X., Han, R., Lv, Q., Mishra, S., Seada, K.: Fusing mobile, sensor, and social data to fully enable context-aware computing. In: Proceedings of the Eleventh Workshop on Mobile Computing Systems Applications HotMobile 2010, p. 60 (2010), http://portal.acm.org/citation.cfm?doid=1734583.1734599
5. Blom, J., Gatica-perez, D., Kiukkonen, N.: People-Centric Mobile Sensing with a Pragmatic Twist: from Behavioral Data Points to Active User Involvement. In: Mobile Devices and Services, pp. 381–384 (2011)
6. Cattuto, C., Broeck, W.V.D., Barrat, A., Colizza, V., Pinton, J.F., Vespignani, A.: Dynamics of person-to-person interactions from distributed RFID sensor networks. PloS One 5(7), 1–9 (2010), http://dx.plos.org/10.1371/journal.pone.0011596

7. Cranshaw, J., Toch, E., Hong, J.: Bridging the gap between physical location and online social networks. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing (2010), http://dl.acm.org/citation.cfm?id=1864380
8. Eagle, N.: Can Serendipity Be Planned? MIT Sloan Management Review 46(1), 10–14 (2004), http://www.angelfire.com/ab8/iissungminshane/smr.pdf
9. Eagle, N., Sandy Pentland, A.: Reality mining: sensing complex social systems. Personal and Ubiquitous Computing 10(4), 255–268 (2005), http://www.springerlink.com/index/10.1007/s00779-005-0046-3
10. Frank, J., Mannor, S., Precup, D.: Generating storylines from sensor data. In: Mobile Data Challenge Workshop (2012)
11. Jonsson, H.: The Data Chicken and Egg Problem. In: Proceedings of the 3rd International Workshop on Research in the Large, pp. 9–12 (2012)
12. Karam, A., Mohamed, N.: Middleware for mobile social networks: A survey. In: 45th Hawaii International Conference on System Sciences Middleware, pp. 1482–1490. IEEE (2012), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber+6149064, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6149064
13. Kiukkonen, N., Blom, J., Dousse, O., Gatica-perez, D., Laurila, J.: Towards rich mobile phone datasets: Lausanne data collection campaign. In: Proceeding of International Conference on Pervasive Services ICPS (2002)
14. Liu, S., Striegel, A.: Accurate Extraction of Face-to-Face Proximity Using Smartphones and Bluetooth. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks, ICCCN, pp. 1–5. IEEE (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6006081
15. Mokhtar, S.B., Mcnamara, L., Capra, L.: A Middleware Service for Pervasive Social Networking. In: Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing, pp. 1–6 (2009)
16. Scholz, C., Atzmueller, M., Stumme, G.: New Insights and Methods for Predicting Face-to-Face Contacts. In: 7th Intl. AAAI Conference on Weblogs and Social Media (2013), http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/viewPDFInterstitial/6097/6396
17. Schuster, D., Rosi, A., Mamei, M., Springer, T., Endler, M., Zambonelli, F.: Pervasive Social Context - Taxonomy and Survey. ACM Transactions on Intelligent Systems and Technology 9(4) (2012)
18. Tran, M.H., Han, J., Colman, A.: Social context: Supporting interaction awareness in ubiquitous environments. In: MobiQuitous, vol. 9, pp. 1–10. IEEE (2009), http://researchbank.swinburne.edu.au/vital/access/services/Download/swin:15475/SOURCE1

# Proximity-based reminders using Bluetooth

H. Jonsson, P. Nugues
Computer Science
Lund University
Lund, Sweden

A. Tavella, I. Amaral, M. Tachibana, V. Santos
Venturus
São Paulo, Brazil

***Abstract*— A smartphone is a personal device and as such usually hosts multiple public user identities such as a phone number, email address, and Facebook account. As each smartphone has a unique Bluetooth MAC address, Bluetooth discovery can be used in combination with the user registration to a server with a Facebook account. This makes it possible to identify a nearby smartphone related to a given Facebook contact. Using this capability, we designed Memorit, an application that handles reminders alerting the user when a contact is nearby her/him. This way it is possible to trigger a user-defined reminder, for example to give back a book, when a registered contact comes in proximity. Data collected from Memorit will allow study of pervasive social context. This paper gives an overview of Memorit, its features, implementation, and evaluates its applicability through a user study.**

***Keywords*— *Bluetooth, Proximity, Reminder, Mobile Device, Social Device, Context***

## I. INTRODUCTION (*Heading 1*)

Bluetooth is a no cost wireless technology that offers short distance transmission. Enabled devices can discover each other in a distance of approximately 10 meters by making themselves discoverable, without the need of a server. Almost all mobile phones shipped in 2012 contained Bluetooth technology [5].

In this paper, we explore Bluetooth device discovery to alert the user about user-defined reminders when a contact mapped to a Bluetooth device is nearby. We describe its implementation in Memorit. We conducted user studies with 13 users to verify the applicability of this concept. The results can be used as guidance in the further development of Bluetooth device interaction with mobile devices.

To the best of our knowledge, no equivalent application of Bluetooth to trigger contact reminders has yet been reported before.

Triggering reminders maybe a simple application, but Memorit is used to investigate the more fundamental consequences of connecting physical identity (through a personal device that is carried around in the form of a telephone) to virtual or digital identities such as a. Facebook identity. The space of applications that are aware of the social context of a user is yet unexplored, although there are calls to establish it as a new domain of research [10].

The rest of this paper is structured as follows. The next section discusses related work. Section 3 describes Memorit. Section 4 explains its implementation. Privacy is discussed in Section 5. Section 6 outlines the methodology used to conduct the user tests, and results. Future work is discussed in Section 7. Finally, the paper closes with the conclusions.

## II. RELATED WORK

Bluetooth proximity detection has been exploited in a couple of previous works. Eagle and Pentland [7] analyzed Bluetooth logs, cell tower distribution, static Bluetooth device distribution, and time of day to discover the relationship among users carrying mobile devices.

In [4], devices implement interfaces and report the Bluetooth signal strength of nearby devices to a centralized server so as to trigger actions on the devices. This way, two nearby devices can, for example, greet each other with sentences such as “Hello, John! How was your vacations?” and “Hello Mary! I had a trip to South Africa.” In our work, there is no need to report the Bluetooth signal strength to the server to have a contact reminder triggered. The Bluetooth connection is established directly from device to device and as soon as one device discovers another one that maps to a contact associated to a reminder, this contact reminder is triggered.

Other applications [6] allow actions in the device to be customized when in Bluetooth proximity to a car, device, notebook, or headset. However, in our work, we focus on social interactions, triggering reminders when a person identified through Facebook or phonebook is in proximity.

## III. THE MEMORIT SYSTEM

The Memorit system consists of a mobile application (Memorit) and a server, Whoownsd, to store the mappings between Bluetooth MAC addresses and Facebook user accounts.

The application registers the user phone number and Facebook account into a server so as to associate the Bluetooth MAC address of the user device with those data. When handling a contact reminder, the application opens a list with all the contacts found in the Android phonebook application and in the user’s Facebook account. The user can then create a reminder for a contact and receive a notification as this contact comes into proximity. Figure 1 shows a screenshot of Memorit with a list of user-defined reminders (left part) and a notification (right part).

If Bluetooth is disabled or if its visibility expires, Memorit will request the user to enable it, since Bluetooth and discoverability is needed to discover and be discoverable by the registered contacts. However, Bluetooth pairing is not needed for the application to recognize that a contact is nearby.

978-1-4799-2736-4/14/$31.00 ©2014 IEEE

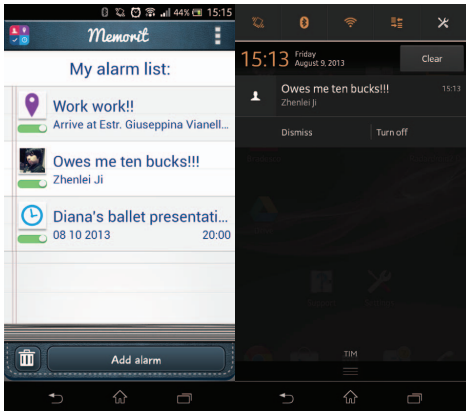



Figure 1. User defined reminders and notification

Apart from handling contact reminders, Memorit also handles time and location reminders in order to be a complete reminder application. Time reminders are triggered based on a selected time and date. Place reminders are triggered based on a location in the map and action (arrive or leave) selected by the user. Figure 2 shows a time reminder (left part) and a place reminder (right part).

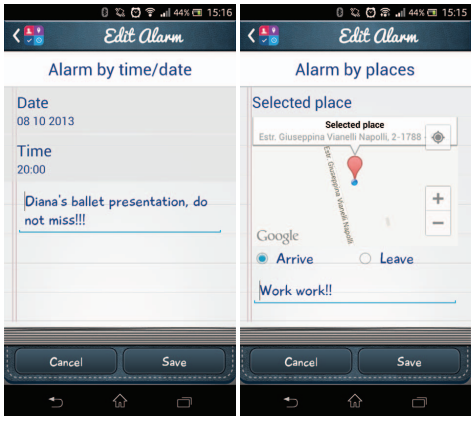



Figure 2. Time and place reminders

## IV. IMPLEMENTATION

Figure 3 shows an overview of the Memorit system. The mobile application uses two standalone components: Proximates and the Whoownsd service. While Whoownsd stores information on devices and their owner's personal user id, Proximates, see Section 4.2, handles device registration, device scanning, synchronization, and broadcast of known contacts.

### A. The Whoownsd Service

Whoownsd is a Java servlet that runs on Google App Engine and maps devices to users/owners. It stores the device Bluetooth MAC address along with its user social network ids, such as Facebook and LinkedIn.

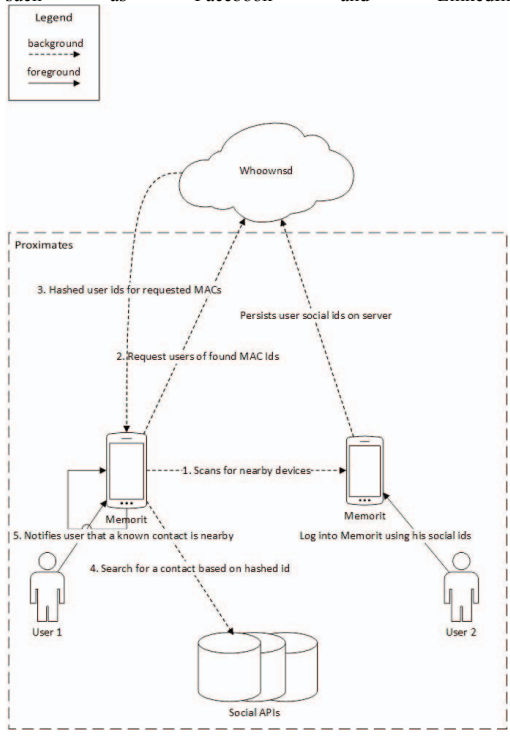



Figure 3. Memorit system overview.

### B. Proximates

Proximates handles the tasks related to proximate users and user id and device registration in Whoownsd.

The registration in the Whoownsd service is done by sending the Bluetooth MAC address of the device and the list of user ids for this device. The user ids registered in Whoownsd are previously hashed using SHA-1 as hash function to preserve the user's identity on the server.

Through the access tokens provided by the Memorit application, Proximates obtains the contacts of the user that will possibly be registered on Whoownsd. These contacts are stored in a local database along with their hashed id, which corresponds to the id returned from Whoownsd for this contact.

Proximates scans periodically for Bluetooth devices, broadcasting whenever a known contact comes in proximity. That is made through the use of a noise control algorithm, which keeps track of the current devices in proximity. In order

to prevent web requests on every scan, since we need the user ids mapped for every Bluetooth MAC, Proximates uses a local cache. The cache is synchronized by a service that runs whenever the user charges his device. This way, the power consumed by the application is considerably reduced.
Proximates also tracks the occurrence of each device and group (set of devices). This way, it is possible to know which devices the user spends most time with. It is also possible to know the relation of the user with a certain group or device (work, friend, family…) through their relationship on the supported social networks. This can be used to infer the user's social context. For example, if s/he is surrounded by work contacts, it is likely that the user is in a work context.
As Proximates is designed as an Android library, third party applications can include it and access its services.

### C. Mobile Application

The Memorit application is available for Android devices on Google Play [12]. It contains the graphic user interface, allowing the user to create, edit, delete, dismiss, turn off reminders, and be notified when the reminder is triggered.
Memorit relies on the Proximates component to be notified when a contact reminder is nearby, and on Google Play Services geofencing [8] to be notified when the device is entering or leaving an area within a given latitude/longitude.

## V. Privacy

Device proximity data is very sensitive data, especially when connecting it to personal identities. In addition to the necessary legal terms of service agreements, we consider informed consent as being a social process and not just a legal transaction [13,2]. To make sure the user stays informed and consenting, the Memorit application was designed to comply with the Obvious Data Usage Principle [2]. The principle states that any data that is used in the application is also reflected in the features of the application in such a way that it is obvious to the user to understand what data is being used.

## VI. Evaluation

Usability evaluation of Memorit application was performed to detect and address inadequacies in the interface. The usability tests were divided in two phases: the first one was performed with 6 users during the initial phase of development. The second stage was performed at the end of development with 6 users to validate the proposed changes suggested in the first phase of testing. The results showed reduction of 29,62% and 77,41% in execution time of the most difficult tasks identified in phase one. According to Nielsen [9], testing with a number of users much greater than 5 does not bring much benefits.

## VII. Future work

With Memorit, we expect to collect a set of proximity data and social network data that allows us to study the temporal dynamics of heterogeneous social networks, both physical and digital. Especially, we will investigate community detection to classify the social context of users in a way that preserves privacy.

## VIII. Conclusion

The usage of Bluetooth discovery to trigger alerts of contacts in proximity has proved to be very useful, as indicated in the user studies conducted for the Memorit system, with users of different sex, age, education, profession, and smartphone usage level.
The need to have Bluetooth in discoverable mode with visibility timeout set to never expire could potentially discourage users to use the Memorit application due to the power consumption. However, Android 4.3 (API level 18) [1] introduced built-in platform support for Bluetooth Low Energy, which aims to provide significantly lower power consumption.
Also, it is already possible to find smart places [4] and meeting rooms equipped with Bluetooth trackers and beacons, which make Bluetooth discovery technique promising.

## Acknowledgment

Our thanks to our users for the participation in the user studies conducted for the Memorit system. This work was partly funded by the Industrial Excellence Center EASE Embedded Applications Software Engineering, (http://ease.cs.lth.se) and the European 7th Framework Program, under grant VENTURI (FP7-288238).

## References

[1] Android developers' website. Bluetooth Low Energy. http://developer.android.com/guide/topics/connectivity/bluetooth-le.html.

[2] Jonsson, H. 2012. The Data Chicken and Egg Problem. In Proceedings of the 3rd International Workshop on Research in the Large. (pp. 9–12).

[3] Garret, J. J. 2002. The Elements of User Experience: User -Centered Design for the Web. United States of America, New Riders Publishing

[4] Mäkitalo, N. et al. 2012. Social devices: collaborative co-located interactions in a mobile cloud. In Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia. ACM, USA.

[5] Bluetooth SIG website: http://www.bluetooth.com/Pages/network-effect.aspx

[6] ToothTag Application,Google Play. https://play.google.com/store/apps/details?id=com.neuaer.toothtag

[7] Eagle, N., Pentland, A., 2005. Social Serendipity: Mobilizing Social Software. In Pervasive Computing, IEEE (May. 2005), 1536-1268.

[8] Android developers' website. Creating and Monitoring Geofences. http://developer.android.com/training/location/geofencing.html

[9] Nielsen, Jakob. 1993. Usability Engineering. Boston: Academic Press.

[10] Yoo, Y. 2010. Computing in everyday life: a call for research on experiential computing. In MIS Quarterly, 34(2), 213–231.

[11] Memorit on Google Play: https://play.google.com/store/apps/details?id=se.lth.cs.memorit

[12] Luger, E., & Rodden, T. 2013. An informed view on consent for UbiComp. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13 (p. 529). New York, New York, USA: ACM Press

2014 IEEE NINTH INTERNATIONAL CONFERENCE ON INTELLIGENT SENSORS, SENSOR NETWORKS AND INFORMATION PROCESSING (ISSNIP)
SYMPOSIUM ON HUMAN-CENTRIC SENSING AND SOCIAL NETWORKS
SINGAPORE, 21–24 APRIL 2014

# A Comparison of Two Proximity Networks

Håkan Jonsson
Lund University
Sweden
Email: hakan@cs.lth.se

Pierre Nugues
Lund University
Sweden
Email: pierre.nugues@cs.lth.se

***Abstract*—We present a comparative exploratory analysis of two proximity networks of mobile phone users, the Proximates network and the Reality Mining network. Data for both networks were collected from mobile phones carried by two groups of users. Periodic Bluetooth scans were performed to detect the proximity of other mobile phones. The Reality Mining project took place in 2004-2005 at MIT, while Proximates took place in Sweden in 2012-2013. We show that the differences in sampling strategy between the two networks has effects on both static and dynamic metrics. We also find that fundamental metrics of the static Proximates network capture social interactions characteristics better than in the static Reality Mining network.**

## I. Introduction

For several years ubiquitous computing used specialized sensor hardware to measure human activities and behaviors [4]. With the increasing penetration and capabilities of mobile phones, especially smart phones, specialized hardware is of less importance. Smartphones allow mobile sensing to scale to thousands of people. A smartphone can be used as a single point of data collection for both sensor and user generated data. From a research point of view, this makes the new field of mobile sensing possible.

Eagle and Pentland introduced the term *Reality Mining* (RM) in a study of 30 office workers using PDAs [6]. However, mobile sensing had its real start only with the collection of the Reality Mining dataset [7]. Eagle and Pentland showed then that mobile sensing could scale to hundreds of users using commodity devices and thus be nonintrusive.

The main difference between the RM and the Proximates datasets is the sampling method: The population of RM is a coherent group of students and employees in the same building at MIT, while the Proximates population was recruited through snowballing, i.e. each participant was asked to recruit more participants. By using snowballing, we believe we can capture richer aspects of users social interactions with less noise than in a study focused on a specific context, for example a campus. In a proximity study in a specific context there is also an increased risk of capturing encounters that are not social interactions. Both the Nodobo [3] study and the SensibleDTU [5] studies are campus based studies like RM, while the Lausanne Data Collection Campaign (LDCC) [9] is a snowballing study. The reason we are using RM rather than LDCC is that the LDCC dataset is not generally available.

In this paper, we provide an exploratory analysis of the data collected in the Proximates project, by comparing its fundamental network metrics to those of the Reality Mining dataset, a dataset well studied.

## II. Data acquisition

We started the data acquisition with the recruitment of users, who were offered to loan high-end mobile phones for the duration of the study, with no other compensation. These users were also asked to recruit friends, family, and colleagues. The study started in July 2012 and ended in October 2013. In total, 176 mobile phones were used in this study on which we installed an application called SmartTodos. SmartTodos is a contextual reminder application that allows users to set reminders based on time, date, when they meet someone, and location. The meeting reminder is triggered by a proximity sensor using Bluetooth. We used SmartTodos to collect proximity and Facebook data that was uploaded to a server and stored for analysis. Bluetooth scans were performed every 2 minutes and we considered only Bluetooth devices with device class "phone". During the study period, our 176 devices collected a total of 4,337,727 Bluetooth scans containing 135,693 unique discovered phones.

For comparison, we used the Reality Mining dataset which was conducted on an equivalent number of users (94) and over a similar time span (September 2004 to June 2005). We also used a Erdösz-Renyi random graph (ER) with the same number of nodes and edges as the Proximates network as a null model for the static analysis.

Each node in the network corresponds to one phone. A link is formed from one device to another when a device is discovered through Bluetooth scanning by another. This allows us to create a directed network. However, in this analysis, we consider the network to be undirected since edges will always be formed by two participants when they are in proximity of each other. The edges are weighted by the number of scans in which the devices appear. Since scans are performed periodically, this translates to time spent in proximity. The range of Bluetooth is up to 10 meters, but in practice it is around 5m. We can thus interpret edges in this network as indicators of social interaction. However, not all edges are social interaction, as for example edges may be created by commuters in proximity on a bus.

There are two main differences between the Proximates and RM networks. The RM network consists of a coherent group of students (26) and employees (68) at MIT, all working and studying in the same building. This makes random encounters during daytime likely. In the Proximates network, users were recruited using snowballing, which means that there is no predefined context that is shared among the participants. The only thing they have in common is that they have been recruited by someone else in the study.


978-1-4799-2843-9/14/$31.00 © 2014 IEEE

In this explorative analysis, we focused on the network consisting of the 176 participants in the study. We did not include the 135,693 discovered phones that have been in proximity of the participants. We also filtered out the nodes with out degree 0.

The data collected in both RM and Proximates have limitations and are subject to a significant noise. Some participants turn off their phones at night, while some do not. Some participants turn off Bluetooth to save battery or install battery savers that prevent background processing, even though they were not allowed to do it during the study. Moreover, Bluetooth is a radio technology, which by its nature is dynamic in interaction with its environment. Thus we should not expect the data collected to completely capture the participants' sensing environment. Also, even if the participants can sense non-participants, we do not capture the participants' complete social network. In Proximates, many participants have only recruited a small portion of their actual social network.

### A. Static analysis

Table I shows a summary of the static metrics extracted from the Proximates, Reality Mining and Erdösz-Renyi networks. We can observe that even though the size of the Proximates network (175) is twice as big as the Reality Mining network (88), the numbers of edges (1241) is less that half of that in the RM network (2946). The median degree in the RM network is more than 6 times higher. The median degree of 70 out of 88 nodes means that almost all nodes are connected to each other. This high connectedness is reflected by Fig. 8 thats shows the RM network structure and Fig. 1 that shows the degree distribution, where 90% of its nodes have a degree higher than 50 and it is only above degree 65 that the frequency starts declining rapidly.

| Network | Proximates | Reality Mining | ER |
|---|---|---|---|
| Nodes | 175 | 88 | 175 |
| Edges | 1241 | 2946 | 1240 |
| Median degree | 11 | 70 | 14 |
| ASP length | 3.09 | 1.23 | 2.21 |
| Diameter | 6 | 2 | 3 |
| Med. Clustering Coeff. | 0.66 | 0.84 | 0.08 |

TABLE I: Summary static metrics

We believe this difference in degree distribution can be explained by the fact that the RM participants are very likely to have been in close proximity to each other at some point in time during the 9 months, even if they did not have any actual social interaction. By just sitting in the same class or passing by each other would create an edge. In contrast, users in the Proximates network are not as likely to be in proximity to each other, unless they actually have a relation and a social interaction.

The local clustering coefficient of a node is computed as the proportion of links between the nodes within its neighborhood divided by the number of links that could possibly exist between them. This is a measure of how connected the neighbors of a node are. The median local clustering coefficients of the Proximates and RM networks are comparable and very different from the ER network. This suggests that even if the median

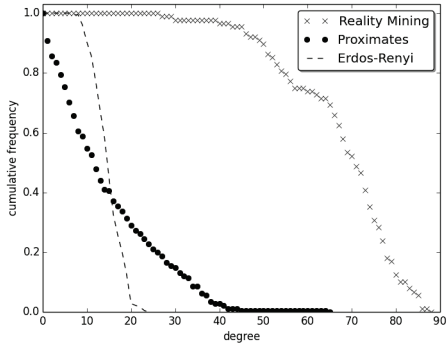



Fig. 1: Complementary cumulative degree

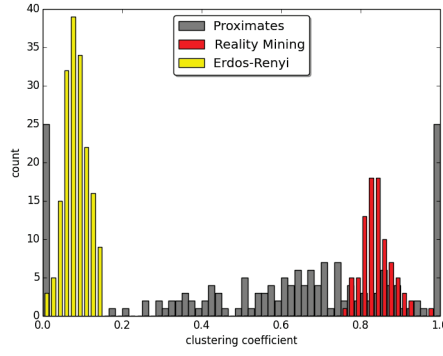



Fig. 2: Clustering coefficient distributions

degree in the Proximates network is low, which is expected in a network using a snowball recruiting strategy: participants are likely to be in proximity of people that have been in proximity of people they have been in proximity of. This indicates that clustering coefficient as a metric captures the spatio-temporal nature of proximity. However, the clustering coefficient distributions (Fig. 2) reveal that the distributions of the ER and RM networks look like normal distributions but with different means, while the Proximates degree distribution is very different from these: A quarter of its nodes has minimum degree, a quarter has the maximum degree, and the rest is spread out in between. The low clustering coefficient nodes are the peripheral nodes with only one neighbor. The number of maximum degree nodes indicates that there is one or more highly connected cliques to which they belong.

The average shortest path (ASP) length and diameter is computed for the largest connected component (LCC) of the Proximates network since the entire nework is not connected. The number of nodes in the LCC is 157 and the number of

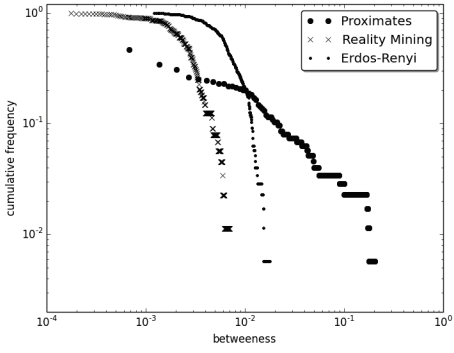



Fig. 3: Complementary cumulative betweenness

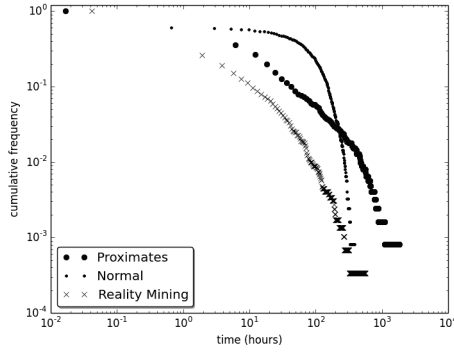



Fig. 4: Complementary cumulative contact time in proximity. For visibility, the graph is cut off at 0.001

edges is 1240. The low ASP length and diameter of the RM network compared to the Proximates network again reflect the fact that the people in the RM network have a high probability of being in proximity at some point even if they have no other relation. In contrast, participants in the Proximates network are less likely to be in proximity to participants they do not know. An ASP of 3.09 for the Proximates network seems to be high in comparison to known networks, e.g. the Facebook friendship network, with an ASP of about 4 [1].

The cumulative betweenness frequencies of the RM and the ER networks (Fig. 3) are similar in shape, both generated by normal distributions, but with different means and standard deviations. The RM network betweenness has a different shape: It seems to have a potential power law region with a sharp cutoff at the tail. This indicates that there are some few nodes with very high betweenness, which is also seen in the network structure visualization. Fig. 9 shows two main clusters with a few nodes connecting them. This is very different from the RM network (Fig. 8) which is made of a single giant densely connected component. Also, since the Proximates network is less densely connected outside the cliques, the betweenness distribution has a fatter tail: More nodes are important when passing information across the network.

Since human proximity is temporal in nature, it makes sense that a static analysis over the whole timespan does not reveal much detail about a network like RM. One temporal aspect is captured in the weight of the edges since they show how much time is spent in proximity between participants, also known as contact time. We compared the RM and Proximates networks with an ER network with weights drawn from a normal distribution (Fig. 4). The distributions of the Proximates and RM networks are similar and different from the ER network. Interestingly, the shape of the distribution of time spent between participants does not seem to be much affected by the structure of the network, especially the degree of the network. This could be explained by the fact that random encounters are more likely to be very short and many, and thus do not contribute to the shape of the tail of the contact time cumulative distributions. Still they contribute as much as any other edge to the degree distributions.

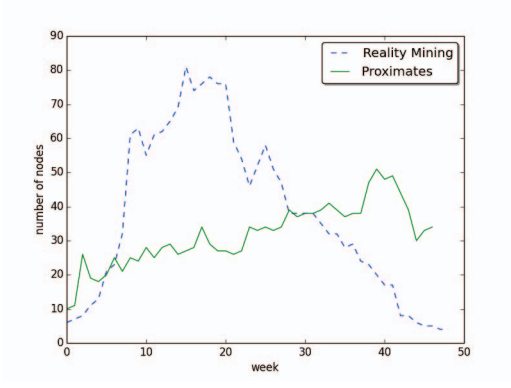



Fig. 5: Number of nodes per week

### B. Dynamic analysis

It is clear that using a snapshot that spans the whole dataset hides a lot of details that would be revealed in a dynamic analysis, taking temporal aspects into account. Here, we provide an initial temporal analysis, where we have chosen to split the dataset into slices, one week long each. Figure 5 shows the number of nodes with an out degree larger than zero per week in the RM and Proximates networks. In Proximates there in an increasing trend throughout the study, as recruiting continues, while the RM network has a drastic peak at around week 15 that lasts for about 10 weeks. The number of edges (Figure 6) in the RM network has a sharp low at week 24, which corresponds to Christmas holidays. No such drastic changes are visible in the Proximates network. The number of edges in the Proximates network is much lower in general. We believe this is due to more spurious interaction, that is

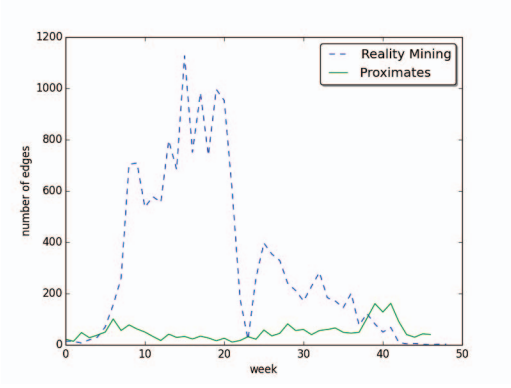



Fig. 6: Number of edges per week

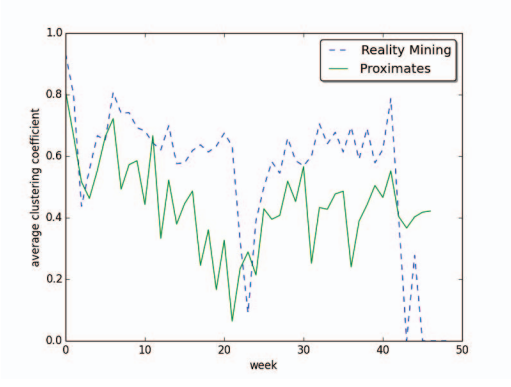



Fig. 7: Average clustering coefficient per week

interactions between participants who do not have a social interaction, but happen to be in the same place. The average clustering coefficients of the networks (Figure 7) measure how many friends of a user meet. Interestingly, they reach a minimum around Christmas holidays for both networks.

## III. Conclusion

In this paper, we have described the static Proximates network, the data collection and its analysis. We showed it captures social interactions to a higher degree than the Reality Mining network, since random encounters that are not social interactions are more likely to occur in the static RM network. We found that degree, betweenness, and clustering coefficient metrics indicate that the interaction processes in RM are stochastic and normally distributed. However, it is also clear that a static analysis over networks that are highly dynamic is only a first step to a complete understanding. Further analysis, especially temporal stability of communities, is suggested as the next step.

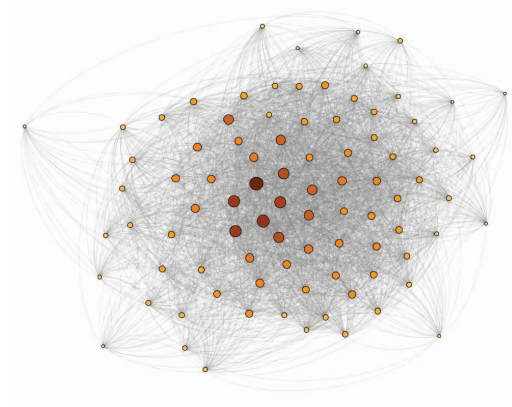


Fig. 8: Reality Mining proximity network. Larger node size and darker color indicate higher betweenness

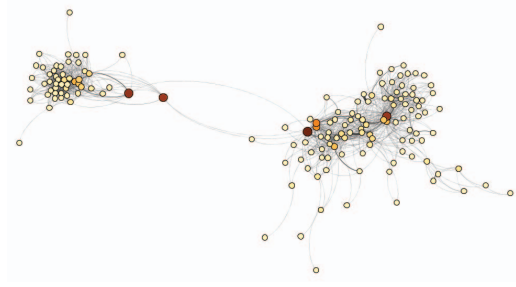


Fig. 9: Proximates proximity network. Larger node size and darker color indicate higher betweenness

## Acknowledgment

The first author would like to thank the authors of Networkx [8] and Gephi [2] graph tools. Our thanks to our users for the participation in the user studies conducted using the SmartTodos application. This work was partly funded by the Industrial Excellence Center EASE Embedded Applications Software Engineering, (http://ease.cs.lth.se) and the European 7th Framework Program, under grant VENTURI (FP7-288238).

## References

[1] L. Backstrom, P. Boldi, and M. Rosa. Four degrees of separation. In *Proc. 4th ACM Int'l Conf. on Web Science (WebSci)*, 2012.

[2] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An Open Source Software for Exploring and Manipulating Networks. *International AAAI Conference on Weblogs and Social Media*, pages 361–362, 2009.

[3] S. Bell, A. Mcdiarmid, and J. Irvine. Nodobo : Mobile Phone as a Software Sensor for Social Network Research. *Electrical Engineering*, 2011.

[4] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn. The Rise of People-Centric Sensing. *IEEE Internet Computing*, 12(4):12–21, 2008.

[5] A. Cuttone, S. Lehmann, and J. Larsen. A mobile personal informatics system with interactive visualizations of mobility and social interactions. In *1st ACM Workshop on Personal Data Meets Distributed Multimedia*, pages 27–30, 2013.

[6] N. Eagle and A. S. Pentland. Social Network Computing. *October*, (October):289–296, 2003.

[7] N. Eagle and A. Sandy Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2005.

[8] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, volume 836, pages 11—-15, 2008.

[9] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-perez, and J. Laurila. Towards rich mobile phone datasets : Lausanne data collection campaign. *Proceeding of International Conference on Pervasive Services ICPS*, 2002.

PLOS ONE

RESEARCH ARTICLE

# SensibleSleep: A Bayesian Model for Learning Sleep Patterns from Smartphone Events

**Andrea Cuttone[1☯]*, Per Bækgaard[1☯], Vedran Sekara[1,3], Håkan Jonsson[3], Jakob Eg Larsen[1], Sune Lehmann[1,2]**

**1** DTU Compute, Technical University of Denmark, Kgs. Lyngby, Denmark, **2** The Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark, **3** Sony Mobile, Nya Vattentornet, Mobilvägen, Lund, Sweden

☯ These authors contributed equally to this work.
* ancu@dtu.dk





**Citation:** Cuttone A, Bækgaard P, Sekara V, Jonsson H, Larsen JE, Lehmann S (2017) SensibleSleep: A Bayesian Model for Learning Sleep Patterns from Smartphone Events. PLoS ONE 12(1): e0169901. doi:10.1371/journal.pone.0169901

**Editor:** Wei-Xing Zhou, East China University of Science and Technology, CHINA

**Received:** August 12, 2016

**Accepted:** December 23, 2016

**Published:** January 11, 2017



**Data Availability Statement:** The research is based on two separate datasets. The data from the SensibleDTU (Dataset B in the manuscript) is attached as Supplementary Information. The data from Sony (Dataset A in the manuscript) is owned by Sony and cannot be made available for download for legal reasons. Access to this dataset can be obtained by contacting Jonas Sellergren (Jonas.Sellergren@sonymobile.com).

**Funding:** AC is funded in part by the High Resolution Networks project (The Villum

## Abstract

We propose a Bayesian model for extracting sleep patterns from smartphone events. Our method is able to identify individuals' daily sleep periods and their evolution over time, and provides an estimation of the probability of sleep and wake transitions. The model is fitted to more than 400 participants from two different datasets, and we verify the results against ground truth from dedicated armband sleep trackers. We show that the model is able to produce reliable sleep estimates with an accuracy of 0.89, both at the individual and at the collective level. Moreover the Bayesian model is able to quantify uncertainty and encode prior knowledge about sleep patterns. Compared with existing smartphone-based systems, our method requires only screen on/off events, and is therefore much less intrusive in terms of privacy and more battery-efficient.

## Introduction

Sleep is an important part of life, and quality of sleep has a significant impact on individual well-being and performance. This calls for methods to analyze sleep patterns in large populations, preferably without laborious or invasive consequences, as people typically disapprove of the use of intrusive technologies [1].

Large scale studies of human sleep patterns are typically carried out using questionnaires, a method that is known to be unreliable. It is possible to perform more accurate studies, but these are currently carried out within small controlled environments, such as sleep labs. In order to perform accurate measurements of sleep in large populations—consisting of thousands of individuals—without dramatically increasing costs, alternative methods are needed.

Smartphones have become excellent proxies for studies of human behavior [2, 3], as they are able to automatically log data from built-in sensors (GPS, Bluetooth, WiFi) and on usage patterns (phone calls, SMS and screen interaction), from which underlying user behavioral patterns can be derived.

Smartphone data has been used to infer facets of human behavior such as social interactions [4], communication [5], mobility [6], depression [7] and also sleep patterns [8]. Either paired

Foundation), as well as Social Fabric (University of Copenhagen). PB is supported in part by the Innovation Fund Denmark through the project Eye Tracking for Mobile Devices. Sony Mobile provided support in the form of salaries for authors VS and HJ, but did not have any additional role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. The specific roles of these authors are articulated in the 'author contributions' section.

**Competing Interests:** The commercial affiliation of VS and HJ to Sony Mobile does not alter our adherence to PLOS ONE policies on sharing data and materials.

with additional sensors or on their own, mobile app solutions are able—sometimes very ingeniously—to track individual sleep patterns and visualize them. We cite as examples *Smart Alarm Clock* [9], *Sleep Cycle* [10], *SleepBot* [11], and *Sleep as Android* [12].

In this paper we suggest extending previous approaches, using a Bayesian model to infer rest and wake periods based on smartphone screen activity information. The advantages of our proposed Bayesian approach *SensibleSleep*, as compared to previous work, are that it:

- is less sensitive to "noisy" data, for instance infrequent phone usage during sleep interruptions (such as checking the phone at night)
- is able to quantify not only specific rest and wake times but also characterize their distributions and thus uncertainty
- can encode specific prior beliefs, for instance on expected rest periods (when desirable)
- can capture complex dependencies between model variables, and possibly even detect and relate patterns that are common to a group of people with diverging individual patterns (when using one of the proposed hierarchical models), such as detecting how available daylight may modulate sleep patterns across an otherwise heterogeneous group of users

Our method, moreover, only needs screen on/off events and is thus *non-intrusive*, *privacy-preserving*, and has *lower battery cost* than microphone or accelerometer based ones.

Although dedicated sleep trackers or fitness tracking bands can provide much more precise and fine-grained data on sleep patterns, their adoption is still quite small in comparison with mobile phones. Therefore the data potentially available from smartphones can enable the analysis of sleep patterns at a very large scale, much larger than using data from dedicated sleep tracking devices.

We start by providing an overview of the related work. We then describe the collected data, and introduce the Bayesian model. We compare the model results with ground truth obtained by sleep trackers, and show how the model is able to infer the sleep patterns with high accuracy. Finally we describe the individual and collective sleep patterns inferred from the data.

## Related Work

A key finding by Zhang et al. [13] shows a global prevalence of sleep deprivation in a group of students, partly linked to heavy media usage. In this study sleep patterns are largely deduced from the teachers' perception or based on individual self-reports, lacking more direct measurements.

Corroborating this finding, Orzech et al. [14] report that digital media usage before bedtime is common among university students, and negatively impacts sleep. The findings are based on studies involving self-reports through (online) sleep diaries and digital media surveys, and also lacks more direct measurements of sleep patterns. Additionally, this would make it possible to increase the scale of the experiment and enable the study of larger populations.

Abdullah et al. [8] have previously demonstrated using 9 subjects how a simple rule-based algorithm is able to infer sleep onset, duration and midpoint based on a (filtered) list of screen on-off patterns with the help of previously learned individual corrective terms, and further analyzed behavioral traits of the inferred *circadian rhythm* [15, 16]. The algorithm uses an initial two weeks of data with journal self-reported sleep for learning key corrective terms in order to improve the accuracy and compensate for differences between *actual* sleep and *inferred* nightly rest period. The method has been verified against a daily online sleep journal and results in differences less than 45 minutes of average sleep duration over the entire analysed period. While our proposed Bayesian model, which has been applied to more than 400 users, may be more

complex, it increases the robustness and allows us to better quantify the uncertainties of the inferred resting periods as well as offer the possibility of building more advanced models across heterogeneous groups of users. In particular, our model may better be able to handle short midnight interruptions, which appear to be not uncommon, without any additional filtering.

In contrast to Abdullah et al. using (only) screen on-off events, a fine-grained sleep monitoring by "hearing" and analyzing breathing through the earphone of a smartphone is suggested by Ren et al. [17]. Here six users tested the system over a period of 6 months, demonstrating the feasibility of using smartphones for the purpose of analysing breathing patterns, using a Respiration Monitor Logger as ground truth. Sleep estimates are not directly inferred in this paper, however. This technology is also non-invasive, although it does requires capturing and analyzing large samples of audio data.

*iSleep* [18] proposes detecting sleep patterns by means of a decision tree model, also based on audio features. The system was evaluated with 7 users for a total of 51 days, and shows high accuracy in detecting snoring and coughing as well as sleep periods, but report drops in performance due to ambient noise.

Increasing the number of features, the *Best Effort Sleep model* [19] is based on a linear combination of phone usage, accelerometer, audio, light, and time features using a self-reporting sleep journal, and subsequently achieved a 42 minutes mean error on 8 subjects in a test period of 7 days.

Other work also tries to estimate sleep quality, for example *Intelligent Sleep Stage Mining Service with Smartphones* [20], which uses Conditional Random Fields on a similar set of features trained on 45 subjects over 2 nights, and reports over 65% accuracy of detection of sleep phases, compared to EEG ground truth on 15 test subjects over 2 nights.

*Candy Crushing Your Sleep* [21] uses the longest period of phone usage inactivity as heuristic for sleep, with some ad-hoc rules for merging multiple periods, and proceeds to quantify the sleep quality and to identify aspects of daily life that may affect sleep. The inferred sleep period was however not validated against any ground truth.

The *Sleep Well* framework [22] deploys a Bayesian probabilistic change-point detection, in parallel with an unsupervised classification, of features extracted from accelerometer data, in order to identify fine-grained sleep state transitions. It then uses an active learning process to allow users to incrementally label sleep states, improving accuracy over time. It was evaluated both on existing datasets with clinical ground truth, and on 17 users for 8-10 days with user diary data as ground truth, reaching an average sleep stage classification accuracy approaching 79%.

In comparison, even though sleep quality is not estimated, our non-intrusive model only needs screen on/off events and has been tested on a large user-base, and can suitable for very large-scale deployment.

## Methods

### Data Collection

We have analyzed two datasets in this work.

The first dataset (A) was provided by Sony Mobile, and contains smartphone app launches coupled with sleep tracking data from the SWR10 and SWR30 fitness tracking armbands [23]. For each user we have a set of records containing an anonymized unique user identifier, a timestamp and the unique app package name. Note that the model only uses the app launch timestamp and completely ignores the app identifier, therefore no privacy risks related to app names are present. The sleep tracking data indicates when each user is detected asleep or awake with a granularity of one minute, serving as ground truth that we will compare our

results against. From this dataset we select 126 users that have at least 3 hours of tracked sleep per day, and have between 2 and 4 weeks of contiguously tracked sleep.

The second dataset (B) originates from the *SensibleDTU* project [24], which collected smartphone sensor data for more than 800 students at the Technical University of Denmark. In this dataset we focus on the screen interaction sensor that records whenever the smartphone screen is turned on or off, either by user interaction or by notifications. Each record contains a unique user identifier, a timestamp, and the event type (on or off). From this dataset we select 324 users in November 2013 that have at least 10 events per day, thus filtering out users with gaps in the collected data or with very sparse data. There is on average $\approx 76$ screen-on activations pr. day pr. user in this period.

Data collection for the SensibleDTU dataset was approved by the Danish Data Protection Agency, and written informed consent has been obtained for all study participants. Data collection for the Sony dataset has been approved by the Sony Mobile Logging Board and written informed consent has been obtained for all study participants according to the Sony Mobile Application Terms of Service and the Sony Mobile Privacy Policy.

## Model Assumptions

The underlying assumptions of the model are (1) that the user is in one of two modes: being *awake* or *sleeping*, and (2) that mobile phone usage differs between the two modes. In particular a user will have many screen interactions when awake, and very few or even no interactions when sleeping.

Sleeping is here considered as an extended *resting* period that typically takes place once every 24 hours at roughly similar times, as governed by the users circadian rhythm and influenced by socio-dynamic structures, during which the owner physically rests and/or sleeps. Resting periods, however, might be interrupted by short periods of activity, such as checking the time on the phone or responding to urgent messages. This behavior leads to two different activity levels, which we label $\lambda_{awake}$ and $\lambda_{sleep}$, one for each mode.

If we can deduce when the switchpoint between the two distributions occur during each 24 hour period, we can also infer the time during which the owner is *resting* for the night, and thereby also the period within which sleeping takes place.

Short of using the more invasive EEG or polysomnographic methods, properly differentiating the resting period and actual sleep is difficult; even sleep diaries may easily contain reporting bias or be somewhat inaccurate. To remove self-reporting bias and to study a larger population we have therefore decided on using a motion-based detector (Sony fitness tracking armbands) as ground truth.

If higher accuracy would be required, applying individual corrective terms (i.e. average sleep/rest time differences) learned from an initial period by more accurate means (polysomnography, external observer or possibly a careful user diary) might be possible, similar to what as demonstrated by Abdullah et al. [8].

## Model Structure

Each user is considered independently. We divide time into 24–hour periods starting at 16:00 and ending at 15:59 on the next calendar day, so that the night period and the expected sleep midpoint is in the middle, for convenience. Each day is divided into $n = 24 * 4 = 96$ time bins of size 15 minutes. We count the number of events that start within each time bin, where an event is an app launch for dataset A and a screen-on for dataset B. Information about the duration of the events is purposely discarded, as phone usage typically takes place in short bursts. This is supported by the median duration of screen events in dataset B, which is $\approx 26.5$ seconds.

A reasonable weakly informative prior assumption is that the count of events $k$ in each time bin follows a Poisson distribution (additional comments on this assumption are available in S1 Appendix):

$$P(k) = \text{Poisson}(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

with $\lambda = \lambda_{awake}$ or $\lambda = \lambda_{sleep}$, depending on the mode of the user. It is, furthermore, assumed that the user mode, and consequently the value for $\lambda$, is determined by two switchpoint variables $t_{sleep}$ and $t_{awake}$, both assuming values from 0 to $n$:

$$\lambda = \begin{cases} \lambda_{sleep} & \text{if } t_{sleep} \leq t < t_{awake} \\ \lambda_{awake} & \text{if } t < t_{sleep} \vee t \geq t_{awake} \end{cases}$$

For simplicity, all models assume that $\lambda_{sleep}$ is identical for all days of a given user. It can be expected that users have a very low number of screen events during sleep mode, which is encoded in this prior belief:

$$\lambda_{sleep} \sim \text{Exponential}(10^4)$$

Here Exponential represents the exponential distribution:

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The rate parameter is set to a very large value to encode our prior belief that almost no events should happen during the sleep time.

Fig 1 shows an illustration of the model idea.

We now propose five different models, which differ in the assumptions made on the relation of the rate and sleep/awake time parameters for different days.

## Pooled-Pooled Model: Pooled Times and Rates

The simplest model assumes that for a given user there is a single $\lambda_{awake}$; i.e. the user has very similar phone interaction patterns each day. Also $t_{sleep}$ and $t_{awake}$ are each identical for all days, that is: the user goes to sleep, and wakes up, at the same times each day:

$$\begin{aligned} t_{sleep} &\sim \text{DiscreteUniform}(0, n) \\ t_{wake} &\sim \text{DiscreteUniform}(0, n) \\ \lambda_{awake} &\sim \text{Gamma}(2.5, 1) \end{aligned}$$

Here DiscreteUniform(0, $n$) represents a uniform probability to choose a timebin between 0 and $n = 96$. No additional prior knowledge of $t_{sleep}$ and $t_{awake}$ is assumed; there is equal probability of any bin value. In other words, sleep and awake time are equally probable at any time of the day. The prior for $\lambda_{awake}$ is chosen to represent our prior belief of a reasonable rate of events, specifically with both mean and variance = 2.5 (events/bin) and a longer tail than a normal distribution.

## Independent-Pooled Model: Independent Times

A somewhat more realistic model would assume that each day has independent $t_{sleep}$ and $t_{awake}$ times, while still sharing $\lambda_{awake}$ rates. Therefore in this model there are $t^i_{sleep}$ and $t^i_{awake}$, with

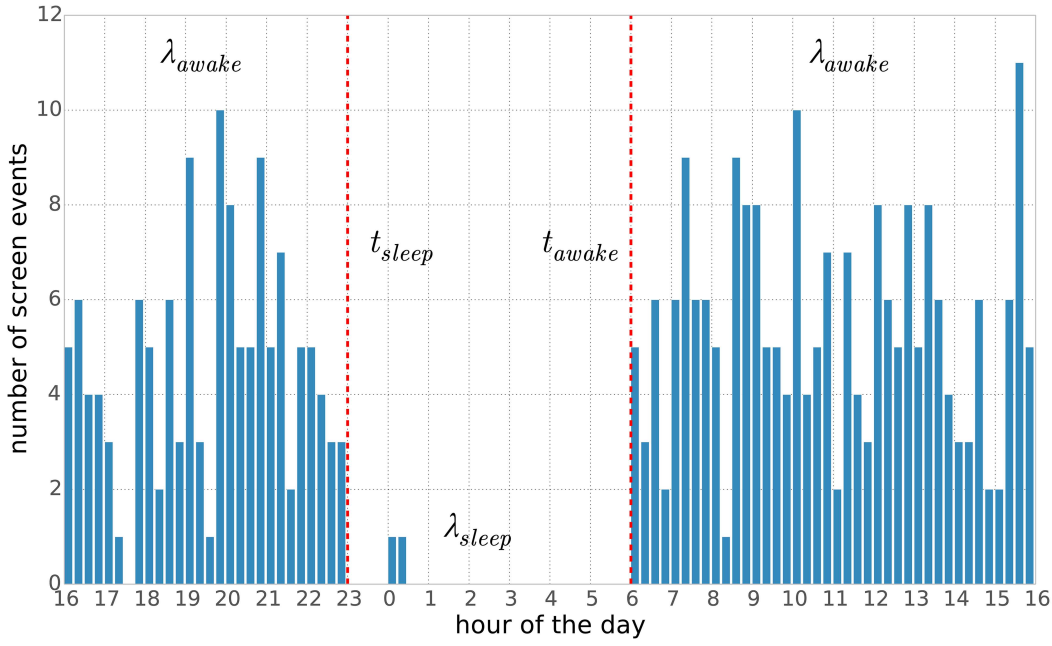



**Fig 1. Conceptual illustration of the model.** We assume that for each day the event counts follow two different Poisson distributions: one for sleep periods (rate $\lambda_{sleep}$) and one for awake periods (rate $\lambda_{awake}$). Furthermore we assume that two switchpoints $t_{sleep}$ and $t_{awake}$ determine the rate (i.e. the Poisson distribution) that generates the events.

doi:10.1371/journal.pone.0169901.g001

$i = 1 \ldots m$, one for the each of the considered days:

$$\begin{aligned} t^i_{sleep} &\sim \text{DiscreteUniform}(0, n) \text{ for } i = 1 \ldots m \\ t^i_{wake} &\sim \text{DiscreteUniform}(0, n) \text{ for } i = 1 \ldots m \\ \lambda_{awake} &\sim \text{Gamma}(2.5, 1) \end{aligned}$$

The rest of the model remains as above.

## Independent-Independent Model: Independent Times and Rates

It may further be assumed that each day could have its own specific activity rate. We modeled this as separate $\lambda^i_{awake}$ for each of the $m$ days, in addition to $t_{sleep}$ and $t_{awake}$ for each of the $m$ days:

$$\begin{aligned} t^i_{sleep} &\sim \text{DiscreteUniform}(0, n) \text{ for } i = 1 \ldots m \\ t^i_{wake} &\sim \text{DiscreteUniform}(0, n) \text{ for } i = 1 \ldots m \\ \lambda^i_{awake} &\sim \text{Gamma}(2.5, 1) \text{ for } i = 1 \ldots m \end{aligned}$$

## Independent-Hyper Model: Hierarchical Rates

The assumption that each day's interaction rate is completely independent may not be correct. It may not be unreasonable to imagine that the daily rate(s) arise from an underlying user-specific rate; i.e. the user may have certain habits that varies from day to day but share some similarities specific to that user. This is modeled by adding $\alpha_\lambda$ and $\beta_\lambda$ hyperparameters to the Gamma priors for $\lambda^i_{awake}$:

$$\begin{aligned} t^i_{sleep} &\sim \text{DiscreteUniform}(0, n) \text{ for } i = 1 \dots m \\ t^i_{wake} &\sim \text{DiscreteUniform}(0, n) \text{ for } i = 1 \dots m \\ \alpha_\lambda &\sim \text{Exponential}(1) \\ \beta_\lambda &\sim \text{Exponential}(1) \\ \lambda^i_{awake} &\sim \text{Gamma}(\alpha_\lambda, \beta_\lambda) \text{ for } i = 1 \dots m \end{aligned}$$

We do not have strong prior beliefs for $\alpha$ and $\beta$, so we set their prior distributions to generic exponential distribution with rate parameter = 1, Exponential(1).

## Hyper-Hyper Model: Hierarchical Times and Rates

Finally we could assume that each day's sleep and awake times derive from an underlying circadian rhythm that is specific to the user, but still modulated by events that take place during the week. This can be modeled by changing the $t^i_{sleep}$ and $t^i_{awake}$ priors to a normal distribution, with hyperparameters $\alpha_t$, $\beta_t$ and $\tau_t$ as follows:

$$\begin{aligned} \alpha_t &\sim \text{Exponential}(1) \\ \beta_t &\sim \text{Exponential}(1) \\ \tau_t &\sim \text{Gamma}(\alpha_t, \beta_t) \\ t^i_{sleep} &\sim \text{Normal}(8 * (n/24), \tau_t) \text{ for } i = 1 \dots m \\ t^i_{wake} &\sim \text{Normal}(15 * (n/24), \tau_t) \text{ for } i = 1 \dots m \\ \alpha_\lambda &\sim \text{Exponential}(1) \\ \beta_\lambda &\sim \text{Exponential}(1) \\ \lambda^i_{awake} &\sim \text{Gamma}(\alpha_\lambda, \beta_\lambda) \text{ for } i = 1 \dots m \end{aligned}$$

The $t^i_{sleep}$ are here chosen to be centered at the bin corresponding to 23:00, while the $t^i_{awake}$ are centered at the bin corresponding to 07:00. Also in this case we have no strong prior knowledge of the $\tau_t$, $\alpha_t$ and $\beta_t$ parameters, so we set their prior distribution to a non-informative Exponential and Gamma respectively.

## Model Fitting and Selection

The models are fitted using Markov Chain Monte Carlo (MCMC) sampling [25], where the parameter values are estimated by a random walk in the parameter space guided by the log likelihood. We use the *pymc3* python library [26, 27] for running the sampling, but any MCMC framework could be used to implement our model. The result of the Bayesian inference is a trace that captures the most probable values of the parameters, and also gives an indication of the uncertainty of the estimation.

It is important to note that the models are unsupervised, which means that they are fitted only to the number of events without having access to the ground truth of the actual sleep

patterns. This allows the model to be fit to other datasets where we do not have ground truth of sleep patterns, which is desirable if the sleep inference has to be deployed on a large scale. For dataset A we verify the fit by comparing with the sleep patterns from sleep trackers, while for dataset B we evaluate the fit by inspecting the inferred sleep patterns.

In order to find the model that provides the best overall fit for the intended purpose without introducing too many degrees of freedom, we compare the log posterior from the traces of the models, logp, and see how they converge.

One example of a plot of logp traces for the five models is shown in Fig 2, which shows that the hyper-hyper model (blue) has the highest (least negative) logp, followed by the independent-hyper model for dataset B. The three other models appear with lower logp. In 76% of the analyzed cases of dataset A (84% for dataset B), the hyper-hyper model has the highest logp score, followed by the independent-hyper model with the highest logp in 11% (13%) of the cases.

The logp estimation does not, however, take into account the added complexity of the more advanced models. An attempt to do so is the Deviance Information Criterion (DIC) [28], which penalizes the increased degrees of freedom (more model parameters) that usually result in a model that is easier to fit to the data. Fig 3 shows the Relative DIC score (vs. the simplest model, pooled-pooled). The order is identical for both datasets.

Further, Table 1 compares the 5 models by ranking the calculated DIC for all 126 and 324 users. The *median* rank shows that the hyper-hyper model is the "best" model; it has a probability of being the best ranked model (*p(Best)*) in 62% of the cases for dataset A (69% for dataset B). The independent-hyper model follows as a somewhat distant 2nd best, ranking highest in 17% (19%) of the cases. For further validation of the goodness of fit of the model, see S2 Appendix.

It should be noted that, in addition to their different abilities to reflect the underlying assumptions and provide varying levels of fit to the actual data, the models also differ in their runtime; the most complex model typically takes 15 times longer to execute than the simplest. In particular, the hyper-hyper model on average had a runtime that is 60% longer than the independent-hyper model, so there may be cases where the latter would be a better model to use despite the slightly worse DIC ranking.

## Results

All five models have been run on both datasets, producing an estimation of the times of sleep and wake up for each day, as well as estimates for the other hyperparameters, for each user. Moreover, we calculated logp and DIC as discussed in the previous section. We firstly verify the accuracy our method using the ground truth from the sleep trackers. We then provide a qualitative analysis of some key examples of individual sleep patterns, and a description of the aggregated sleep patterns for both datasets. For the remainder of the paper we restrict our analysis to the model with the best fit, the hyper-hyper model.

### Comparison to Related Work and to Ground Truth

To assess the results, we compare the sleep periods inferred by our model and those inferred by a previously suggested rule-based method to the ground truth collected by the Sony sleep trackers.

For each day we calculate the time of sleep and time of awake inferred by our model as the mean of the $t^i_{sleep}$ and $t^i_{wake}$ respectively, and we consider the user asleep (Z = 1) for all time bins between $t^i_{sleep}$ and $t^i_{wake}$, and awake (Z = 0) for the remaining bins.

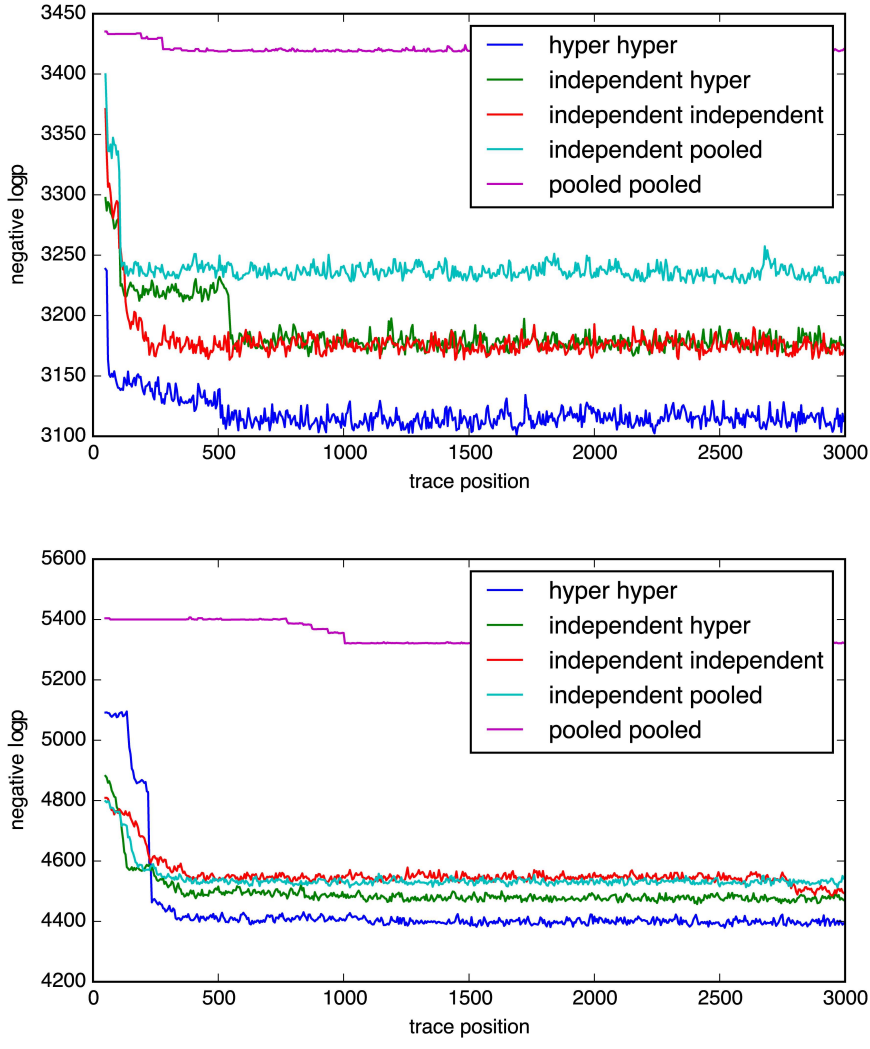



**Fig 2. Typical logp traces (A top, B bottom).**

doi:10.1371/journal.pone.0169901.g002

For a representative and comparable method, we chose to implement a rule-based algorithm similar to what is proposed by Abdullah et. al. [8] to derive sleep data for dataset A. This rule-based method essentially works by finding the longest contiguous sleep period, with a prior assumption that sleep must start after 10 PM and before 7 AM next morning. Note that the original algorithm is based on screen on-off events and furthermore discards events of

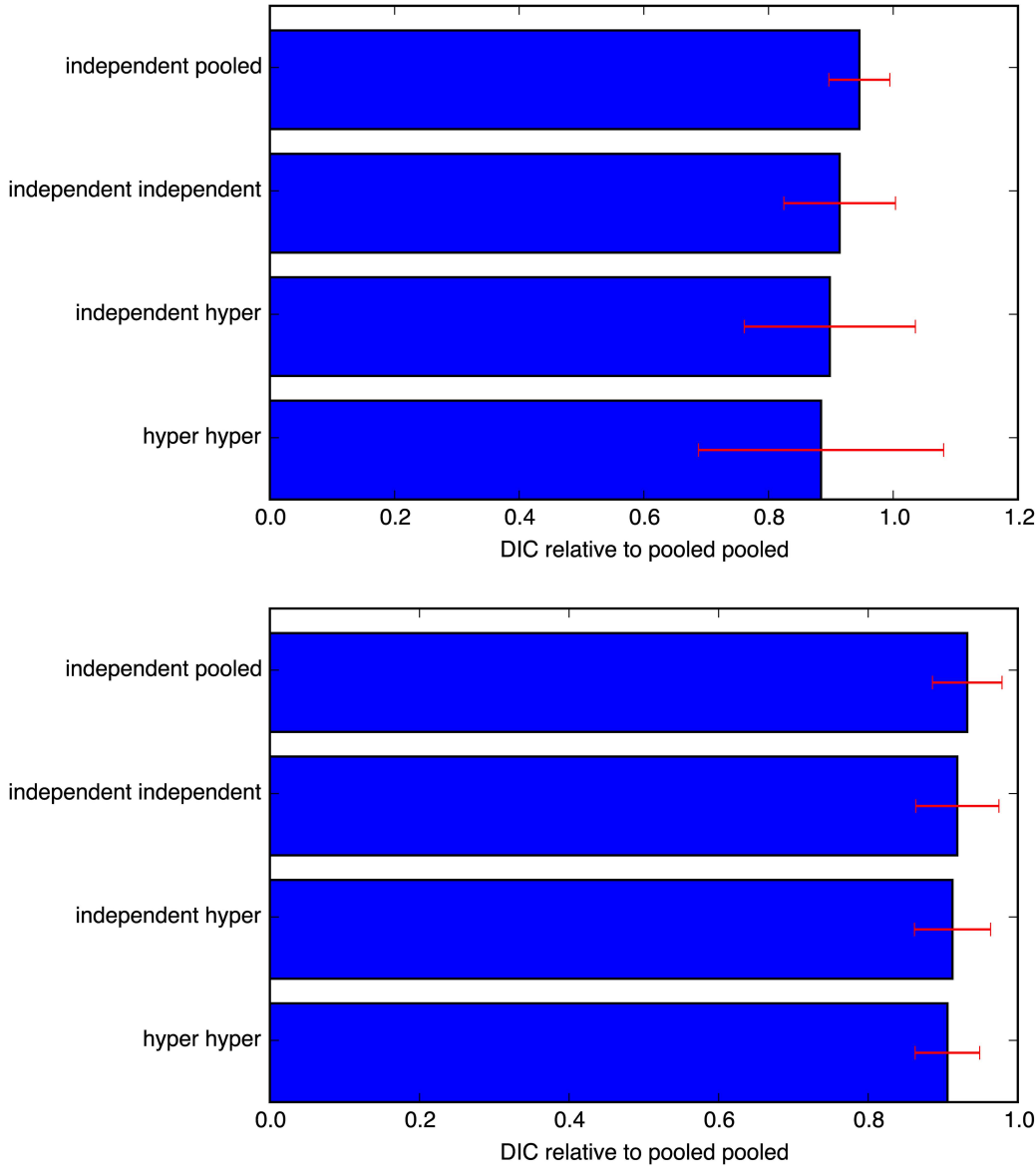



**Fig 3. Relative DIC scores (A top, B bottom), sorted by their mean value (error bars represent one standard deviation).** For both datasets the order is the same, with the hyper-hyper model having the lowest mean DIC.

doi:10.1371/journal.pone.0169901.g003

**Table 1. Model DIC comparisons.**

| | Model Ranks | Median | Mean | | *p(Best)* | Mean Relative DIC | |
|---|---|---|---|---|---|---|---|
| | | | Value | (StdDev) | | Value | (StdDev) |
| A | pooled-pooled | 5 | 4.27 | (1.37) | 0.10 | 0.96 | (0.16) |
| | independent-pooled | 4 | 3.82 | (0.85) | 0.03 | 0.95 | (0.05) |
| | independent-independent | 3 | 2.86 | (1.08) | 0.08 | 0.91 | (0.09) |
| | independent-hyper | 2 | 2.29 | (0.83) | 0.17 | 0.90 | (0.14) |
| | hyper-hyper | 1 | 1.76 | (1.11) | 0.62 | 0.88 | (0.20) |
| B | pooled-pooled | 5 | 4.70 | (0.89) | 0.02 | 0.99 | (0.01) |
| | independent-pooled | 4 | 3.75 | (0.66) | 0.02 | 0.93 | (0.05) |
| | independent-independent | 3 | 2.92 | (1.02) | 0.09 | 0.92 | (0.06) |
| | independent-hyper | 2 | 2.06 | (0.69) | 0.19 | 0.91 | (0.05) |
| | hyper-hyper | 1 | 1.56 | (0.94) | 0.69 | 0.91 | (0.04) |

doi:10.1371/journal.pone.0169901.t001

short duration during the night; in our case we use app launches with no available duration, and thus cannot discard events of short duration.

For the sleep trackers we can directly mark each time bin as sleep (Z = 1) if the trackers have detected at least one sleep status in that bin, and awake (Z = 0) otherwise.

We again consider one user at a time. For each user we now have three binary matrices: two inferred sleep status values per time bin from either model, and one measured sleep status value per time bin (ground truth). We evaluate this as two binary classification problems, and calculate accuracy, precision, recall and F1 for each model and for each user according to the definitions:

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{predictions}}$$

$$\text{precision} = \frac{\text{true positives}}{\text{predicted positives}}$$

$$\text{recall} = \frac{\text{true positives}}{\text{all positives}}$$

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Fig 4 shows the resulting distribution of accuracy, precision, recall and F1 scores for the proposed method. The SensibleSleep method achieves a mean accuracy of 0.89, and a mean F1 score of 0.83. The below-average scores for some users are expected, since it is likely that among the large population under study there will be people having irregular sleep schedule or noisy sleep ground truth.

Fig 5 shows the corresponding *complementary cumulative distributions* of the accuracy, precision, recall and F1 scores of the proposed SensibleSleep model vs that of the rule-based model [8]. The results are generally comparable between the two models, on this particular dataset. Our model has slightly better accuracy and precision whereas the previously suggested rule-based model has a slightly better recall. The F1 scores, which weights precision and recall equally, are comparable.

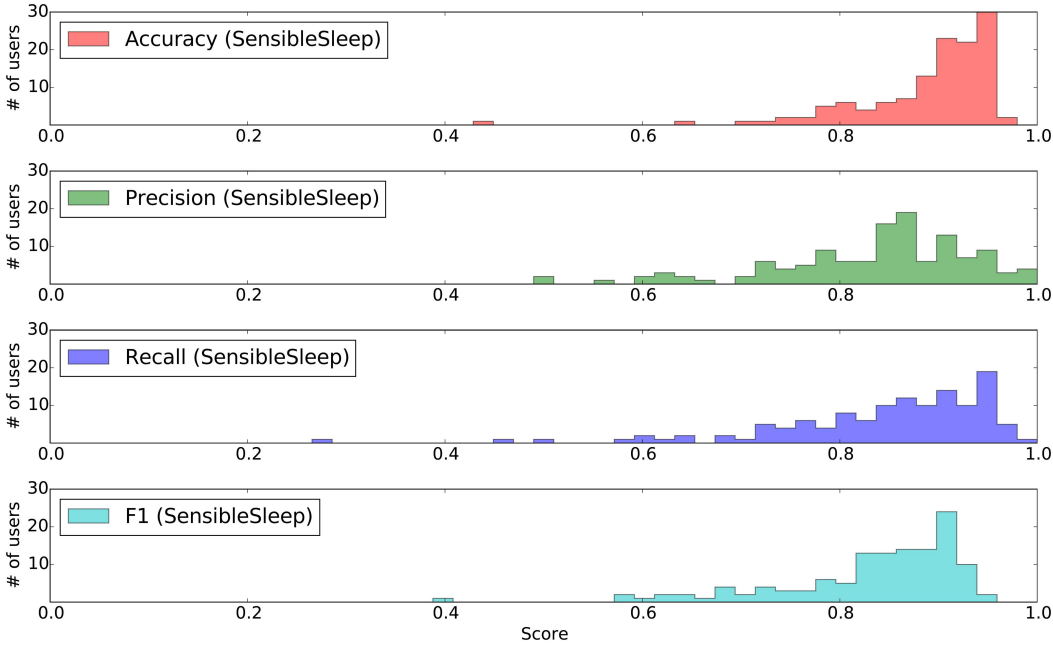



**Fig 4. Histogram of the calculated accuracy, precision, recall and F1 score for users in dataset A, comparing the proposed method to the sleep tracker ground truth.**

doi:10.1371/journal.pone.0169901.g004

## Individual Sleep Patterns

We now analyze individual sleep patterns to show the results of the model in details. For each user we create a visualization of sleep schedules. We call this the *sleep matrix*. Each row represents one day, and each column represents one time bin. The blue color shows the probability that sleep takes place within the interval; the darker the color the higher the probability. The red dots show activity count per bin; the larger the radius the more events are registered within that particular bin. This compact representation is able to capture at a glance the sleep patterns of individuals over time. We have created one such sleep matrix for each of the users, which allows us to inspect hundreds of sleep patterns quickly. Large individual variability both in sleep schedules (regular, irregular) and in phone activity (low, high, during day or night) are noticeable. Still, in most cases it is evident that the model is able to capture a reasonable sleep period, even if it may have been somewhat interrupted.

Let us consider the inferred sleep patterns for two example users in Fig 6. The top user has a pretty regular schedule, waking up around 5:30 except every few days, when he/she wakes up later—presumably due to vacation or weekends. Notice the light blue sections that indicate how the model is less confident about the probability of sleep due to events that do not follow the usual patterns. The bottom user instead has a much more unstable app usage, therefore the model infers a correspondingly more unstable sleep schedule. The bottom user has also some events in the middle of the night throughout many days (which is presumably checking the phone at night) yet the model is still able to correctly infer this being a sleep phase. Finally

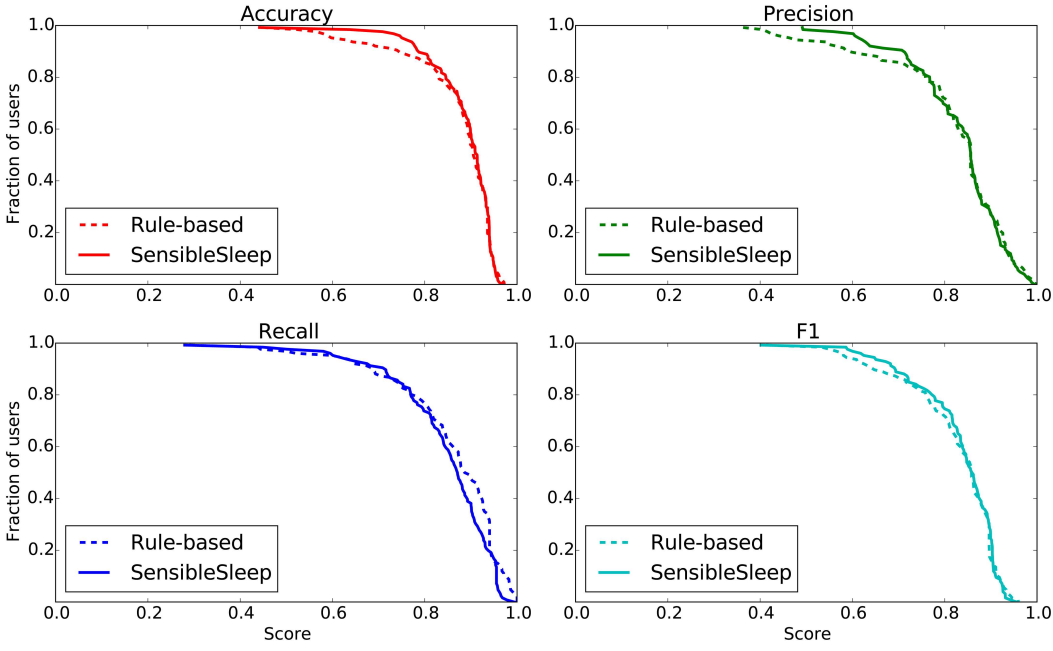



**Fig 5. Complementary cumulative distribution of accuracy, precision, recall and F1 scores for users in dataset A, comparing the proposed model (solid line) to the rule-based model (dashed line), showing the proportion of users (y-axis) having a score less than or equal to a specific value (x-axis).**

doi:10.1371/journal.pone.0169901.g005

notice how the two users have significantly different intensity of app usage (the bottom one uses the phone much more than the top one), yet this is not a problem since the model learns individual activity rates.

## Aggregated Sleep Schedules

In this section we also quantify the aggregated sleep patterns. From the posterior probability distribution functions (PDFs), $P_{t_{sleep}}(t)$ and $P_{t_{awake}}(t)$, the probability that the user is sleeping can be estimated as follows:

$$P_{sleep}(t) = P_{t_{sleep}}(t) - P_{t_{awake}}(t)$$

This is equivalent to stating that a user is currently sleeping if he has passed the time of falling asleep but has not yet passed the time waking up.

The derived values of sleep-length $t_{sleeplength}$ and mid-sleep time $t_{midsleep}$ can be calculated directly from the values of $t_{sleep}$ and $t_{awake}$ for each sample of the trace, and the posterior density can be estimated for these derived values in a similar way as for the model parameters. Fig 7 shows the aggregate posterior probability density functions for $t_{sleep}$ and $t_{awake}$ for the 126 users of dataset A over 15–30 days, and for the 324 users of dataset B over a selected period of 30 days (just after semester start).

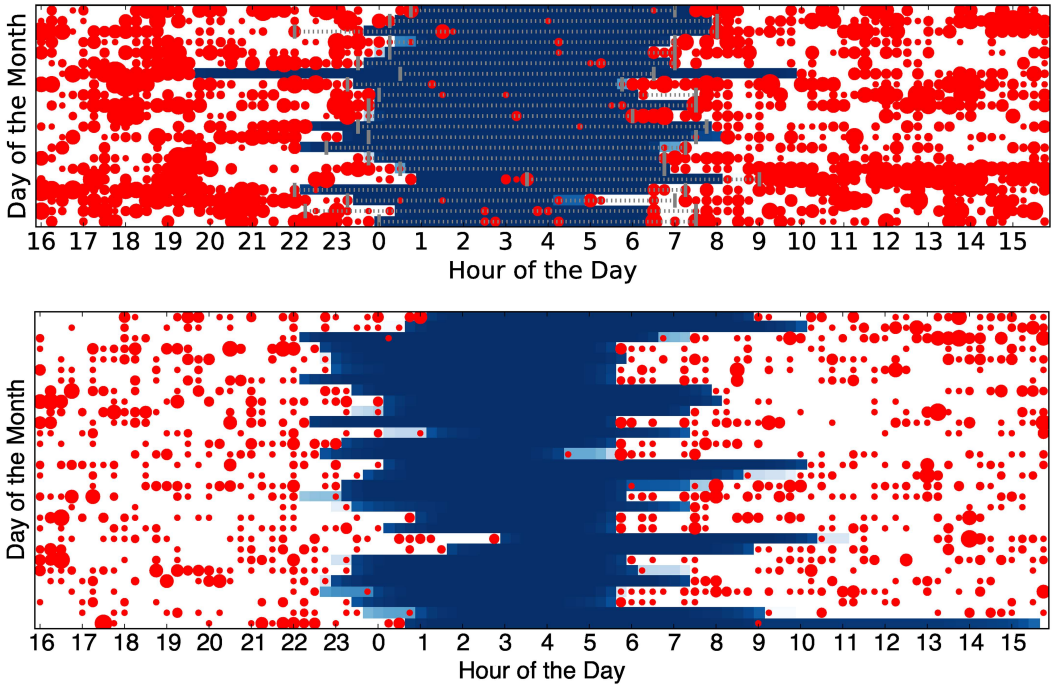



**Fig 6. Sleep matrix of two sample users (21 days from dataset A top, 30 days from dataset B bottom).** The ground truth derived from the sleep tracker is shown as a dotted line overlaid the matrix for dataset A.

doi:10.1371/journal.pone.0169901.g006

It may not be entirely meaningful to average the sleep patterns from all users, but it serves to illustrate the distribution of $t_{sleep}$ and $t_{awake}$ for a larger population. Table 2 summarizes the sleep and wake times.

Across the 30 (14-28) analyzed days for the 324 (126) users of the study, the distribution of sleep durations are as shown in Fig 8. The model allows us to easily compute such metrics. The mean value is around 8:02 (±2h 36m) for dataset A and 7:20 (±2h 28m) for dataset B. Notice how the distributions are not completely similar; this is likely due to the fact that the larger dataset B captures the sleeping behavior of students as opposed to dataset A that may have a more diverse demographic distribution.

Fig 9 shows the probability density functions for the $t_{sleep}$ and $t_{awake}$ times for all users of dataset B, grouped according to weekday. Mondays to Thursdays appear quite similar, but Friday shows a much wider distribution; users typically go to bed much later on Friday and sleep in on Saturday. The distributions start to narrow down Saturday and Sunday but are more "week-like" only from Tuesday morning again.

## Discussion

The main contribution of this work is to show how simple counts of smartphone interactions can be used to infer sleep patterns with reasonably high accuracy. We have demonstrated how

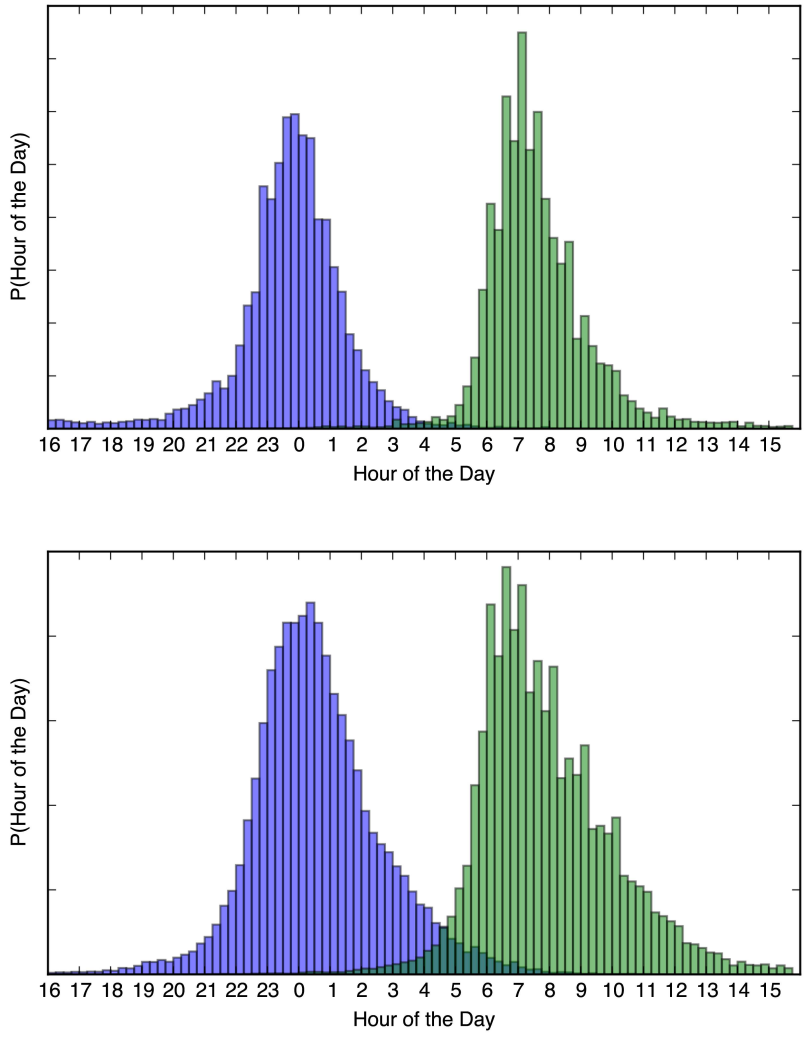



**Fig 7. Aggregate Posterior Probability Distributions of $t_{sleep}$ (blue) and $t_{awake}$ (green) (A top, B bottom), showing what the probability is for the specific population to go to sleep or wake up at the specified time.**

doi:10.1371/journal.pone.0169901.g007

the seemingly weak signal of screen events carry significant information of the user status. Our method has several advantages:

- The method requires only a smartphone and can therefore be deployed without the need for special equipment or methods, such as fitness or sleep tracking bands, or sleep diaries.

**Table 2. Aggregated sleep and wake times.**

| | Sleep Time | | Wake Time | |
|---|---|---|---|---|
| | Mean | (Std) | Mean | (Std) |
| A | 23:38 | (2h 16m) | 7:40 | (2h 2m) |
| B | 0:35 | (2h 6m) | 7:55 | (2h 15m) |

doi:10.1371/journal.pone.0169901.t002

- The data collection is completely automated, as no action is required from the user in setting up the tracking or remembering to log his/her activity.
- Since the model requires only screen interactions, it is absolutely non-intrusive and privacy-preserving. Although in this work we stored the data on a central server for analysis purposes, the data could remain on the phones and the sleep analysis could in principle be run directly on the phones as well.
- Compared to accelerometer or microphone-based methods, using only screen events is much more battery-efficient.

Although solutions using screen events have been proposed before [8, 21], our model provides a number of key improvements:

- It is more robust to noise such as screen events generated by checking the phone at night.
- Using a Bayesian formulation allows us to provide confidence intervals for the sleep and awake times, instead of point estimates only.
- It does not depend on ad-hoc rules, but it is based on a well-defined statistical formulation.
- It is fitted and verified on a much larger userbase of over 400 users, and a longer time duration (between 2 and 4 weeks).

Demonstrating the feasibility of inferring reasonable sleep patterns from simple event counts opens the way for new exciting research directions. In particular we believe that similar methods can be applied to large datasets of user activity. For example on social network (such as Twitter, Facebook, Meetup, Gowalla) users leave a trace of their activity in the form of messages, posts, likes, etc. Another great example is Call Detail Records, the logging information kept by telecom providers about user calls and SMS. These events could be treated again as a proxy for sleep and wake cycles.

The main drawback of the proposed method is that it requires that users periodically interact with their phones during their wake time. In line with other recent polls (see for example [29–31]), we show that in most cases this does happen, as the population of users analyzed here tend to check their phone from the early morning to the late night when awake. Different populations, however, such as elderly people less accustomed to smartphone usage, may not show similar usage patterns. There is therefore a need for additional work in order to understand how increased sparsity would affect sleep pattern reconstruction.

## Conclusions

We have presented a Bayesian model to infer sleep patterns from smartphone interactions, which we have applied to two datasets of more than 400 users in total. We have compared the model output with ground truth from sleep trackers, and we have shown how the model is able to recover the sleep state with a mean accuracy of 0.89 and a mean F1 score of 0.83. Furthermore, we have shown how the model is capable of producing very reasonable individual

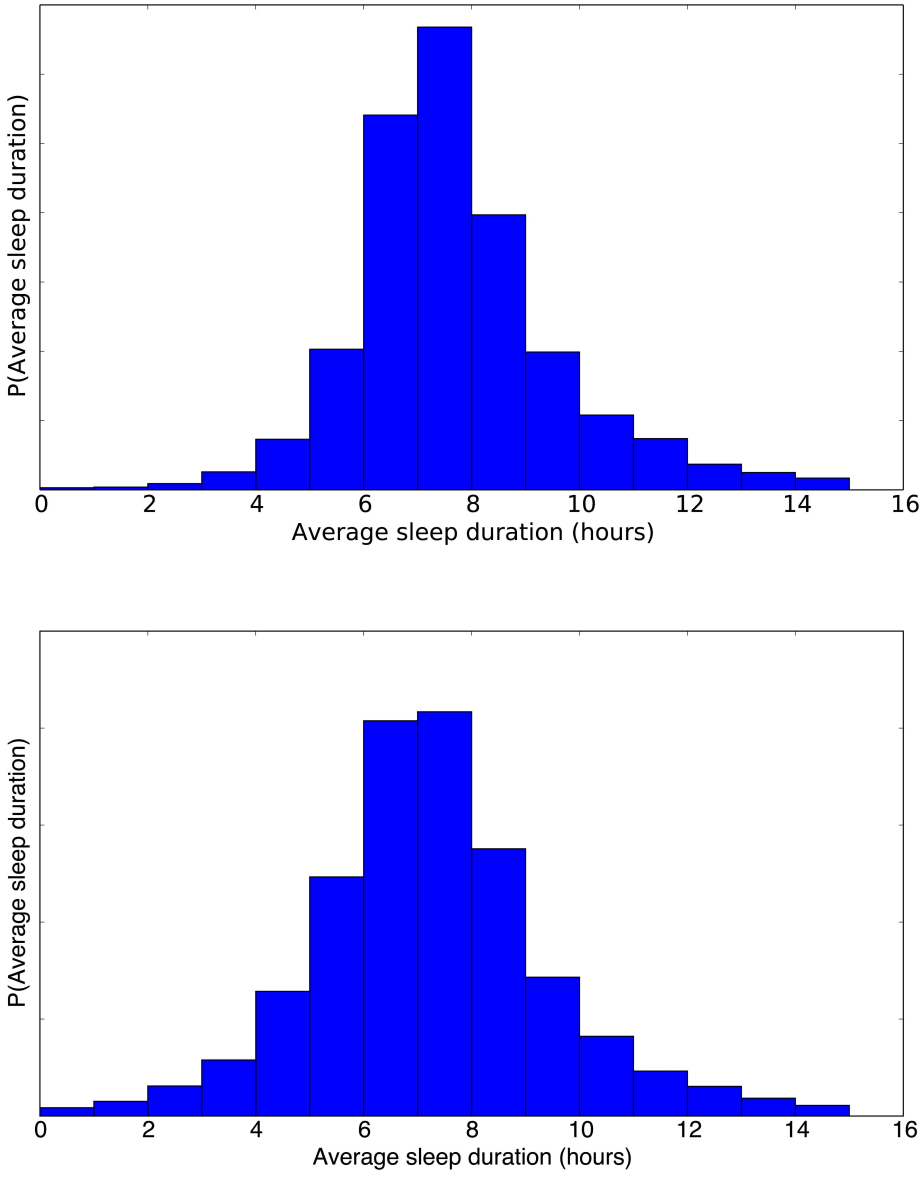



**Fig 8. Aggregated Sleep Durations (A top, B bottom), based on the Posterior Probability Functions.** This illustrates the probability of the length of a nights sleep within the population within the datasets.

doi:10.1371/journal.pone.0169901.g008

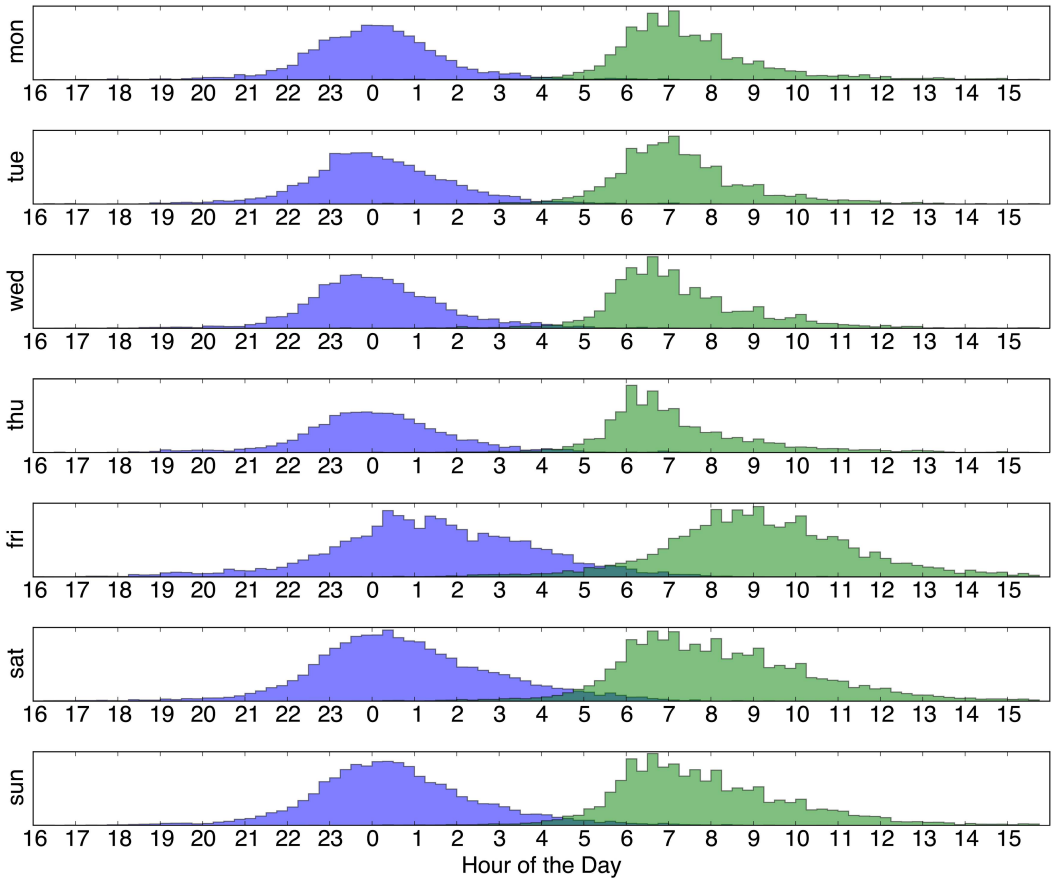



**Fig 9. $t_{sleep}$ (blue) and $t_{awake}$ (green) over weekdays for dataset B.**

doi:10.1371/journal.pone.0169901.g009

and aggregated sleep patterns. Our method represents a cost-effective, non-intrusive and automatic alternative for inferring sleep patterns, and can pave the way for large-scale studies of sleep rhythms.

## Supporting Information

**S1 Dataset. Screen interaction events from SensibleDTU.** The screen-on events including Unix timestamp and user id.
(ZIP)

**S1 Appendix. Poisson vs Power Law Distribution for screen events.**
(PDF)

**S2 Appendix. Goodness of Fit.**
(PDF)

## Author Contributions

**Conceptualization:** AC PB HJ VS JL SL.

**Data curation:** AC PB.

**Investigation:** AC PB.

**Methodology:** AC PB HJ VS JL SL.

**Software:** AC PB.

**Validation:** AC PB.

**Writing – original draft:** AC PB HJ VS JL SL.

**Writing – review & editing:** AC PB HJ VS JL SL.

## References


1. Choe EK, Consolvo S, Watson NF, Kientz JA. Opportunities for Computing Technologies to Support Healthy Sleep Behaviors. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI'11. New York, NY, USA: ACM; 2011. p. 3053–3062. Available from: http://doi.acm.org/10.1145/1978942.1979395.
2. Candia J, González MC, Wang P, Schoenharl T, Madey G, Barabási AL. Uncovering individual and collective human dynamics from mobile phone records. Journal of Physics A: Mathematical and Theoretical. 2008; 41(22):1–11. doi: 10.1088/1751-8113/41/22/224015
3. González MC, Hidalgo CA, Barabási AL. Understanding individual human mobility patterns. Nature. 2008; 453(7196):779–782. doi: 10.1038/nature06958 PMID: 18528393
4. Sekara V, Stopczynski A, Lehmann S. The fundamental structures of dynamic social networks. arXiv preprint arXiv:150604704. 2015;.
5. Onnela JP, Saramäki J, Hyvönen J, Szabó G, Lazer D, Kaski K, et al. Structure and tie strengths in mobile communication networks. Proceedings of the National Academy of Sciences of the United States of America. 2007; 104(18):7332–7336. doi: 10.1073/pnas.0610245104 PMID: 17456605
6. Zheng Y, Zhang L, Xie X, Ma WY. Mining interesting locations and travel sequences from GPS trajectories. Proceedings of the 18th international conference on World wide web—WWW'09. 2009;(49):791.
7. Saeb S, Zhang M, Karr CJ, Schueller SM, Corden ME, Kording KP, et al. Mobile Phone Sensor Correlates of Depressive Symptom Severity in Daily-Life Behavior: An Exploratory Study. Journal of Medical Internet Research. 2015; 17(7):e175. doi: 10.2196/jmir.4273 PMID: 26180009
8. Abdullah S, Matthews M, Murnane EL, Gay G. Towards Circadian Computing: "Early to Bed and Early to Rise"Makes Some of Us Unhealthy and Sleep Deprived. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. 2014; p. 673–684.
9. Smart Alarm Clock;. https://itunes.apple.com/us/app/smart-alarm-clock-sleep-cycles/id586910133?mt=8.
10. Sleep Cycle;. https://itunes.apple.com/us/app/sleep-cycle-alarm-clock/id320606217?mt=8.
11. SleepBot;. https://play.google.com/store/apps/details?id=com.lslk.sleepbot&hl=en.
12. Sleep as Android;. https://play.google.com/store/apps/details?id=com.urbandroid.sleep&hl=en.
13. Zhang M, Tillman DA, An SA. Global prevalence of sleep deprivation in students and heavy media use. Education and Information Technologies. 2015; p. 1–16.
14. Orzech KM, Grandner MA, Roane BM, Carskadon MA. Digital media use in the 2 h before bedtime is associated with sleep variables in university students. Computers in Human Behavior. 2016; 55:43–50. doi: 10.1016/j.chb.2015.08.049
15. Richter CP. Biological Clocks in Medicine and Psychiatry: Shock-Phase Hypothesis. Proceedings of the National Academy of Sciences of the United States of America. 1960; 46:1506–1530. doi: 10.1073/pnas.46.11.1506 PMID: 16590778

16. Aschoff J. Circadian Rhythms in Man. Science (New York, NY). 1965; 148(3676):1427–1432. doi: 10.1126/science.148.3676.1427
17. Ren Y, Wang C, Yang J, Chen Y. Fine-grained sleep monitoring: Hearing your breathing with smartphones. In: Computer Communications (INFOCOM), 2015 IEEE Conference on. IEEE; 2015. p. 1194–1202.
18. Hao T, Xing G, Zhou G. iSleep: unobtrusive sleep quality monitoring using smartphones. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems. ACM; 2013. p. 4.
19. Chen Z, Lin M, Chen F, Lane ND, Cardone G, Wang R, et al. Unobtrusive sleep monitoring using smartphones. In: 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth). IEEE; 2013. p. 145–152. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6563918.
20. Gu W, Yang Z, Shangguan L, Sun W, Jin K, Liu Y. Intelligent Sleep Stage Mining Service with Smartphones. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. UbiComp'14. New York, NY, USA: ACM; 2014. p. 649–660. Available from: http://doi.acm.org/10.1145/2632048.2632084.
21. Jayarajah K, Radhakrishnan M, Hoi S, Misra A. Candy Crushing Your Sleep. In: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers. UbiComp/ISWC'15 Adjunct. New York, NY, USA: ACM; 2015. p. 753–762. Available from: http://doi.acm.org/10.1145/2800835.2804393.
22. Hossain HMS, Roy N, Khan MAAH. Sleep Well: A Sound Sleep Monitoring Framework for Community Scaling. In: Proceedings of the 2015 16th IEEE International Conference on Mobile Data Management —Volume 01. MDM'15. Washington, DC, USA: IEEE Computer Society; 2015. p. 44–53. Available from: http://dx.doi.org/10.1109/MDM.2015.42.
23. Sony SmartWear;. http://www.sonymobile.com/global-en/products/smartwear/.
24. Stopczynski A, Sekara V, Sapiezynski P, Cuttone A, Madsen MM, Larsen JE, et al. Measuring large-scale social networks with high resolution. PloS one. 2014; 9(4):e95978. doi: 10.1371/journal.pone.0095978 PMID: 24770359
25. Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian data analysis. vol. 2. Taylor & Francis; 2014.
26. Patil A, Huard D, Fonnesbeck CJ. PyMC: Bayesian stochastic modelling in Python. Journal of statistical software. 2010; 35(4):1. doi: 10.18637/jss.v035.i04 PMID: 21603108
27. Fonnesbeck CJ. PyMC version 3; 2015. Available from: https://github.com/pymc-devs/pymc3.
28. Berg A, Meyer R, Yu J. Deviance information criterion for comparing stochastic volatility models. Journal of Business & Economic Statistics. 2004; 22(1):107–120. doi: 10.1198/073500103288619430
29. Tecmark survey finds average user picks up their smartphone 221 times a day; 2014. http://www.tecmark.co.uk/smartphone-usage-data-uk-2014/.
30. Newport F. Most U.S. Smartphone Owners Check Phone at Least Hourly; 2015. http://www.gallup.com/poll/184046/smartphone-owners-check-phone-least-hourly.aspx.
31. BankOfAmerica. Trends in Consumer Mobility Report; 2015. http://newsroom.bankofamerica.com/files/doc_library/additional/2015_BAC_Trends_in_Consumer_Mobility_Report.pdf.

# Group affiliation detection in a challenging environment

1st Hakan Jonsson
*Research and Incubation*
*Sony Mobile Communications*
Lund, Sweden
hakan.1.jonsson@sony.com

2nd Pierre Nugues
*Computer Science*
*Lund University*
Lund, Sweden
pierre.nugues@cs.lth.se

***Abstract*—Social interaction sensing and indoor positioning using are widely researched. However, many use cases only need to determine proximity, and not the exact location. In this paper, we describe two methods to determine which meeting each user is participating in using proximity data collected from a challenging real-world office. We show that the RSSI threshold approach to detecting proximity is not feasible due to the optimal RSSI range being very small and close to pessimal ranges. Instead, we achieve an F-score of 82% with a simple method, $k$-nearest neighbor classification, using data from the whole population. This method does not need any historic data or training, calibration to environment, nor find a specific RSSI threshold. Finally, we present result from a user study with a prototype meeting application that identifies meeting participants, and advice on consequences of the above result for UI design.**

***Index Terms*—Peer-to-peer computing, social computing**

## I. Introduction

Indoor positioning of people using Wifi or Bluetooth is a popular research topic. Existing solutions often require infrastructure such as WiFi or Bluetooth base stations placed throughout the area in which positioning is needed, with costs and complexity as a consequence. A possible solution is to use the GPS positioning capabilities of mobile phones, carried by most people, but these methods are not robust in indoor environments. However, there are many use cases for indoor positioning that actually don't need an exact position, but only the proximity of people to other people. One such use case is meeting participation detection, i.e. to determine which people are together in an office meeting room during a meeting. A method to discover proximity between devices is to use Bluetooth, which is available on most mobile phones. This allows us to determine the proximity between people, assuming they carry mobile phones. Compared to using positioning, this method does not require any fixed infrastructure or communication.

Bluetooth has been used in two previous studies, which attempted to identify face-to-face interactions between mobile phone users, using an estimated distance based on Bluetooth RSSI, and an optimal threshold value to filter out non participating users [1]–[3]. They both find very different values, indicating that this is a hard problem and that there are factors influencing this approach that was not taken into account by these studies.

In this paper, we describe two methods for determining meeting participation using proximity data collected from real meeting room environments. In the first method, we apply a filter to the Bluetooth RSSI data collected locally on a phone. We report an optimal value for RSSI threshold that is surprisingly low as well as findings regarding time window size that contradict previous results [1]. We show that the RSSI threshold approach to detecting proximity is not feasible due to the optimal RSSI range being very small and close to pessimal ranges. The second method consists in applying a $K$-nearest neighbor classification to RSSI data collected from all nearby phones, for which we achieve an F-score of 82%. This method allows us to identify interactions without having to find a specific RSSI threshold. Finally, we describe an application that uses this technique to identify the ongoing face-to-face interactions a user is participating in.

### A. Related work

[1] designed a probabilistic framework inspired by Latent Dirichlet Allocation to mine human interactions types from proximity data. They also evaluate the quality of using Bluetooth to sense social interactions, using a weekly meeting calendar event as ground truth. This is probalby not a very good source of ground truth, since it is not actually known who is present and not. [2] study the relationship between the value of Bluetooth RSSI and face-to-face interaction distance between two users, measuring both indoor and outdoor conditions. RSSI and distance based on empirical measurements. In order to reduce error rates to acceptable levels, they also needed to include light sensors data into their model, which is not feasible for many use cases. They did not study detection of groups. [3] validate the use of Bluetooth RSSI as proxy for social interaction detection, measuring RSSI at different device distances in a lab setting as well as in the real world. They then show the effect of different signal strength threshold values and time scales on link filtering in the social network. The RSSI and distance measurements were done in a very controlled lab environment, and do not necessarily generalize to challenging environments.

[4] use a threshold approach with Wifi for detecting group affiliations in various settings, including real office environments. This method requires calibration including knowing

the number of walls between sender and receiver. They do not report on power consumption, but the Wifi scanning rate of 10s used indicates a very high power consumption. Also, the distance error reported (2m for 95th percentile error) is not enough for our challenging environment since the attendees are closer than that even when in different rooms (Figure 1).

[5] use a combination of Wifi and Bluetooth in a peer-to-peer system to determine distances between peers. This system does not require any server. However, the reported distance accuracy (2,7m for 90th percentile error) is not enough for our use case. Also the power-saving mechanism used would not save much power in a challenging office environment where many changes would require a high scanning rate.

[6] design a system using Wifi and evaluate both fingerprinting and signal strengths approaches to flock detection and users membership in them. The system is evaluated in a real world office environment with impressive results. This work is the one closest to ours. The main difference is that the distances in the office environment used for evaluating the system [6] are much larger, probably making it easier to separate flocks. Also, the office was not an densely populated open landscape which most modern offices are, like the one we used.

## II. Experiments and Evaluation

### A. Problem description

Previous studies have tried to identify face-to-face interactions using an estimated distance based on Bluetooth RSSI and an optimal threshold value to filter out non participating users [1]–[3]. We set up an experiment in a real office environment to determine if these results apply to face-to-face interactions in the form of meetings in neighboring meeting rooms. The filter is used to classify each discovered device during Bluetooth scans as being present or not present in the same meeting as each classifying device.

We also tried to identify the meeting a device is participating in and the room it is in. With this problem formulation, we only need to classify the label of the meeting or meeting room, rather than trying to classify each discovered device as a participant or not.

Our assumption is that both approaches can be used to estimate which face-to-face meeting a user carrying a phone is participating in, if any.

### B. Experimental setup

We placed 11 phones, 6 Xperia Active and 5 Xperia Mini, in four adjacent meeting rooms of an office. Figure 1 shows the configuration. The phones were placed on meeting room tables, in positions where participants in a meeting would put them if they would put their phones on the table. Phone orientation was varied across the devices. During 30 minutes, Bluetooth scans were collected every minute from each device. Each scan contained a time stamp, the MAC IDs of the scanning device and the detected devices, and the RSSI values for each of them. The six meeting rooms were placed in the middle of an open office landscape. During the data collection 25 additional devices of various models, outside the meeting rooms, some stationary and some not, were detected.

### C. RSSI threshold filtering

We analyzed the scanned RSSI values from each phone and we applied an RSSI threshold filter to each scan sample. Scanned devices that had an RSSI value lower than the threshold were classified as being not present in the same meeting as the device that performed the scan, while those higher or equal were classified as being present. This was compared to the ground truth presence of devices in different meetings rooms, as shown in Figure 1. From this, we calculated the average precision and recall figures across all the samples and devices. Additionally, we calculated the same figures on multiple scans for different numbers of scans. When using multiple scans, all the devices discovered in a scan are added to a set of discovered devices from the other scans within the time window. In Figure 2, the precision values are marked by solid lines, and the recall values by dotted lines.

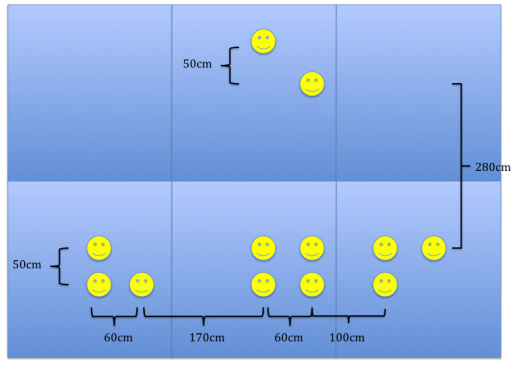



Fig. 1. Meeting rooms layout and participant distances

Figure 2 shows that we have a perfect recall for all RSSI values up to -30dB, after which it drops sharply to below 0.5 depending on number of scans included. The precision grows for increasing RSSI values up to -32dB where it reaches 0.67. The top solid precision line denotes the values calculated for a single scan. It is not surprising that we get the best precision from just a single scan rather than multiple scans: Since we have perfect recall up to -30dB, adding additional scans can only add potential false positives.

The conclusion from this analysis is that choosing an RSSI threshold of -32dB and using a single scan results in the best threshold when determining which devices are in the same meeting room. This is a higher value than previous studies [2], [3] have found. However, the RSSI value for a maximum precision (-32dB) is only 2dB from the value where recall drops drastically (-30dB). Considering the fluctuations in RSSI over time, a filter should use a lower value than -32dB to avoid affecting recall.

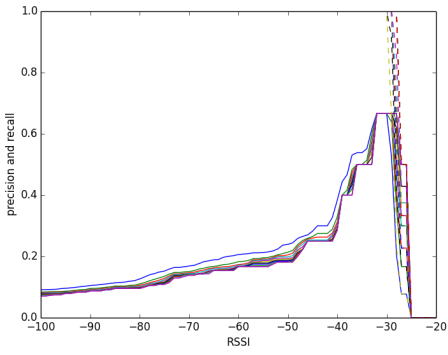



Fig. 2. Precision and recall of detected participants vs RSSI

### D. Global nearest neighbor classification

The second method we applied uses global information, i.e. the signal strength measurements from all phones. In this case, each sample is a 36-dimensional vector containing the measurements of all nearby phones as measured by each phone. The i-th element in the vector represents the value measured by the i-th device. The 36 dimensions include the 11 phones collecting the data, but also all the discovered nearby devices in the office that were not participating in any of the four meetings and not collecting data. They are still included in the data since part of the problem is to be able to handle the noise they contribute. We label each sample with the meeting the device that collected the sample, is participating in.

For each device, we applied a $k$-nearest neighbor classification using the data collected from all the other phones, to predict the label of the meeting the device is participating in. The number of neighbors was chosen to four, corresponding to an estimated average number of participants in a meeting in the office we used. Cross validation was used to estimate the performance of the classification, with random sampling, and assuming time independence of samples. The precision, averaged over all the devices, was 83% while the recall was 81%, resulting in an F-score of 82

The main advantage of using $k$-nearest neighbor classification is that we do not need to find a specific RSSI threshold value. A specific threshold value is likely to create a brittle filter since it is dependent on devices used and environment. Also, the optimal RSSI range seems very small and close to pessimal RSSI value (Figure 2). With $k$-nearest neighbors, we only look at what devices are close to each other in the space spanned by the current environment and devices, using well defined similarity measures.

## III. Application

We prototyped an application that helps the user take notes during meetings. One feature of the application is to detect the presence of people in the meeting, i.e. its attendees 3. The user taking notes can compare the list of invitees and detected attendees and manually correct any errors. To detect the identity of the owner of a present device, the Proximates [7] middleware is used. Proximates maps Bluetooth MAC IDs to social identities, for example Facebook. Proximates also provides with Bluetooth scanning and logging functionalities. Thus, it allows the application developer to focus on the device owners instead of the devices.

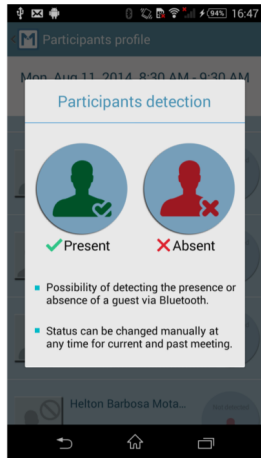



Fig. 3. Application user interface

In the prototype, we used the local filtering method. In a future work, we will include the $k$-nearest neighbor version too, in order to evaluate them on actual users. The $k$-nearest neighbor method requires the use of a server, to which each client uploads RSSI and calendar data (meeting identifier). The server applies a classification and returns the resulting meeting identifier to each device. It can also share the list of discovered attendees in each meeting to the other participants. Security issues and procedures are of course very significant in order to not reveal sensitive meeting or participant information.

We performed a qualitative user study, interviewing 12 users of the application to understand the usefulness and performance of the attendee detection feature. All users found the feature useful. Regarding performance, it was very clear that attendee detection must optimize for recall rather than precision, since it is much easier to delete a false positive (a user who was erroneously added to the attendee list) than to add an false negative (an attendee erroneously not detected).

## IV. Limitations

The collected data set is limited in the sense that we used a single device model; the number of participants in meeting is only between two and four; the meeting rooms are all the same size; they all have the same construction; and the sampling was done only in the morning. To further validate the results,

we will collect additional data from more meetings and in different environments, at different times, and with a diverse set of devices.

## V. Conclusion

Based on the differences in results from different studies, we conclude that there is a huge variance in Bluetooth RSSI between environments and devices that previous studies do not take into consideration. The differences we observe across studies make it futile to search for a single RSSI threshold value to use for general face-to-face group affiliation detection. Even using different values for different environments [2] does not cover the differences observed across studies. For studies using the same device model for all participants, for example SensibleDTU [8], the problem of differences between devices can partly be ignored by providing all the participants with the same device. However, for real world applications, this approach is not feasible. Our first results using global RSSI information with a $k$-nearest neighbor classification are promising. The approach allows us to ignore the absolute values of the signal strengths and instead focus on other similarity measures in a multidimensional space. It is a simple method that does not require fitting to historic data or calibration to an environment.

This was just a small study i a single environment, and several more studies must be done to validate and generalize results.

## Acknowledgments

This work was partly funded by the Industrial Excellence Center EASE Embedded Applications Software Engineering, (http://ease.cs.lth.se).

## References

[1] T. Do and D. Gatica-Perez, "Human interaction discovery in smartphone proximity networks," *Personal and Ubiquitous Computing*, 2013. [Online]. Available: http://link.springer.com/article/10.1007/s00779-011-0489-7

[2] S. Liu and A. Striegel, "Accurate Extraction of Face-to-Face Proximity Using Smartphones and Bluetooth," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, Jul. 2011, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6006081 http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6006081

[3] V. Sekara, S. Lehmann, and C. Science, "Application of network properties and signal strength to identify face-to-face links in an electronic dataset," pp. 1–11, 2013.

[4] A. Matic, V. Osmani, and O. Mayora-Ibarra, "Analysis of social interactions through mobile phones," *Mobile Networks and Applications*, vol. 17, no. 6, pp. 808–819, 2012.

[5] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, "Virtual compass: relative positioning to sense mobile social interactions," in *International Conference on Pervasive Computing*. Springer, 2010, pp. 1–21.

[6] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster, "Mobile sensing of pedestrian flocks in indoor environments using wifi signals," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*. IEEE, 2012, pp. 95–102.

[7] H. Jonsson and P. Nugues, "Proximates–a social context engine," in *Evolving Ambient Intelligence*. Springer International Publishing, 2013, pp. 230–239.

[8] "SensibleDTU," https://www.sensible.dtu.dk, accessed: 2014-03-12.

# User privacy attitudes regarding proximity sensing

Håkan Jonsson
Lund University
hakan.jonsson@cs.lth.se

Carl Magnus Olsson
Malmö University
carl.magnus.olsson@mau.se

## ABSTRACT

User attitudes on privacy with respect to location data has been extensively studied. However, user attitudes of privacy in relation to proximity sensing is still lacking. We present the results from a survey conducted on users of a proximity sensing application we developed and diffused by handing out 100 phones with the proximity sensing application pre-installed. The results are compare this type of application to location sensing in general, as well as positions our respondents in relation to previous studies in terms of general privacy policies. Four results stand out in particular: One, our respondents are considerably more aware of and care about privacy policies than in previous studies. Two, trust is reported as being based more on the specific data access asked for, than EULA or similar text based policies. Third, the respondents are willing to share privacy related data as long as they are in control of who can access it. Finally, our results indicate that there is no perceived difference in sensitivity between location and proximity sensing.

## CCS CONCEPTS

• **Security and privacy → Social aspects of security and privacy**;

## KEYWORDS

Privacy, user attitudes, proximity, Bluetooth

**ACM Reference format:**
Håkan Jonsson and Carl Magnus Olsson. 2018. User privacy attitudes regarding proximity sensing. In *Proceedings of 1st Interdisciplinary Workshop on Privacy and Truse, Hamburg, Germany, August 2018 (iPAT 2018)*, 5 pages.
https://doi.org/10.475/123_4

## 1 INTRODUCTION

There have been several studies on sharing and privacy aspects on location data, for example Sadeh et al. [2]. A likely reason for this is that location sharing services are widely used. There is, however, a surprising lack of reflection on privacy aspects related to proximity. Addressing this is becoming increasingly relevant e.g. by the growing diffusion of indoor positioning services that rely on user-proximity rather than GPS coordinates. Still, there seems to be an assumption that sharing proximity data may be more sensitive from a privacy perspective. From a technical standpoint, there have also been numerous reports on Bluetooth security vulnerabilities (e.g. Tan and Sagala Aguilar [4]), which negatively impact the exploration of proximity-based services. However, upon reviewing these reports, we were unable to find any research reporting on user perceptions of privacy with respect to Bluetooth visibility or other forms of proximity sensing. This suggests that the present assumptions need to be scrutinized rather than assumed as de facto truth.

To this end, we developed and diffused a proximity based service and conducted a survey with active users of this service. For the purposes of this paper, our main focus is on the user study in regards to their attitudes to proximity information. This was broken down to trust in three actor categories (friends, application developers, and service providers), and three aspects (identity, location and proximity information) related to these. As a control question, to position our respondents in relation to previous studies, general questions on end user license agreements (EULA) were used. As our respondents indicate significantly higher care and awareness of such privacy policies, the responses we received to our specific research interests are particularly likely to be after careful consideration. This indicates that the validity of our results is strengthened, thereby increasing the potential for further research to rely on and expand upon our study.

The remainder of the paper sets off by presenting the proximity based service we designed and diffused, starting with the component base to then move on to the end-user service. After this, we present the survey setup and its results in a reflection oriented section where we also position the results in relation to previous studies. We end the paper with a brief summary of findings, and acknowledgments for the support in this work.

## 2 PROXIMITY BASED SERVICE

Given the relative scarcity of proximity based end-user services, ensuring user familiarity has been a long-term goal of our research. The design of our proximity based service therefore relies upon a component based approach, where each component has been developed as a stand-alone component to allow multiple end-user services to be defined depending on the purpose of the inquiry. For this paper, a proximity based reminder service called Memorit was developed and diffused.

### 2.1 Component base

Two stand-alone components, Proximates and a Whoownsd service, form the base upon which the end-user service Memorit was designed. The Proximates component negotiates all proximity tasks, including users, user id and device registration. Whoownsd is a Java servlet running on the Google App Engine, mapping devices to users, which stores the Bluetooth MAC address together with relevant other user id:s (such as e.g. Facebook and LinkedIn). These additional user id:s are needed to establish social proximity, as our notion of proximity is based on more than physical distance.


*iPAT 2018, August 2018, Hamburg, Germany*

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

Periodically, Proximate scans for devices broadcasting over Bluetooth to identify when a Proximate user is within proximity, using a noice control algorithm to pick relevant signals out of all current Bluetooth devices that are responding to the broadcast. To circumvent the need to use web requests upon every scan, as user id mappings with Bluetooth MAC addresses is needed, we rely on a local cache. This cache is synchronized using a service which only runs as the user charges the device, thereby reducing the power consumption of the application significantly. Furthermore, Proximate tracks devices and sets of devices a single user has, as they occur in proximity to another Proximate device user. This makes it possible to identify which device a user may spend most time with, as well as the relation this user may have with another device (or set of devices). From a user perspective, this - together with the supported social network services - Proximate can know relationships between users (colleague, friend, family member, etc) to infer social setting for an interaction. For instance, if the user is surrounded by colleagues, it is likely that the setting is work related, which may be of relevance to an end-user service design.

## 2.2 The Memorit end-user service

In the case of Memorit, this end-user service is a proximity-based reminder. It may thus cut across multiple different user categories, with some reminders being relevant when meeting colleagues, some when meeting a particular friend, while others are relevant for family members. As a user sets up the application, phone number and social media account - passed on from Memorit via Proximate to the Whoownsd servlet - is used to map the MAC address of the registered devices with those data. The user may then create reminders for when a particular contact of theirs is in proximity. These reminders show up as notifications with the user-defined content. On the left of Figure 1 is an example of such a reminder list, while on the right is the notification details.

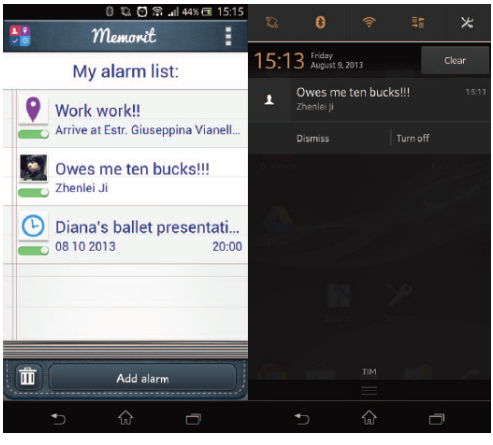



**Figure 1: User defined proximity reminders**

Aside from the proximity to other users they have some form of relationship with, Memorit also allow traditional time based reminders (Figure 2, left) and reminders based on place (right) where notifications may be triggered upon arriving or leaving (for example triggering a notification to remember picking up groceries on your way from work). An overview of the Memorit system, and the components it is based on is available in Figure 3.

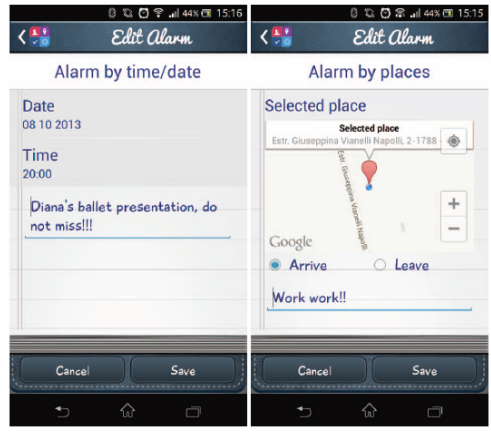



**Figure 2: Time and place based reminders**

## 3 SURVEY RESULTS AND REFLECTIONS

The Memorit application was published on Google Play (under a different name) with over 1000 users at the time of conducting this study. In addition, more than 100 users were recruited using snowballing to act as principal user group for inquiry. Snowballing implies that each user of an initially small set of users, was asked to recruit additional users, and so on. This approach was selected to ensure practical aspects such as living within the same region and naturally occurring proximity interaction between users. The principal inquiry users received phones with Memorit pre-installed to ensure that they all were up and running, and understood the concept.

The survey was distributed by text messages to the Memorit principal inquiry group and 31 users responded to the survey. The age distribution of users is shown in Figure 4. The minimum age was 14, and all users below the age of 18 had written permission of their parents to participate in the study.As we wanted to understand how our respondents compared with other user groups, we included questions similar to what other studies have observed in application and location based privacy surveys. After a general question on privacy policies, we narrowed down our questions to trust in three actor categories (friends, application developers, and service providers) and to three privacy aspects (identity, location and proximity) related to these actors.

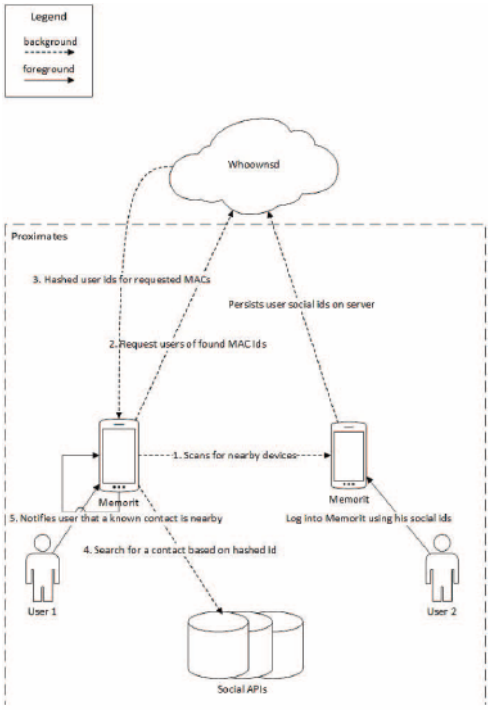



**Figure 3: Overview of the Memorit system and base components**

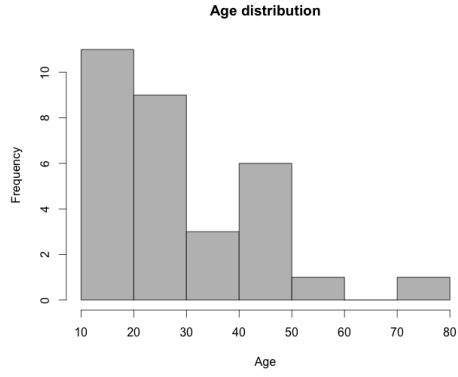



**Figure 4: Age distribution of respondents**

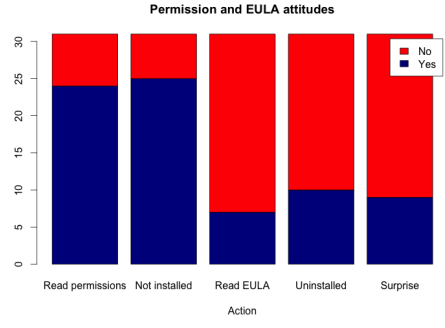



**Figure 5: Permission and EULA attitudes**

## 3.1 Privacy policies

Application stores like Google Play can be modelled as asymmetric markets where application sellers have more information about data collection and processing performed by the application than the buyer [5]. One information signal in Google Play is the permission request shown to the user when installing an app. To understand how important this signal is to users, we asked if they read permission requests before installing, if they have ever decided to not install an app due to permission requests, whether they read end user license agreements (EULA) or privacy policies, and if they have ever decided to uninstall an application due to its EULA or policy. Finally we asked if they had ever been surprised by what data an application uses or publishes after installation.

As can be seen in Figure 5, most users read permission requests but not EULAs. The number of users who read permissions (77%) was much higher in our study than what was found in Felt et al. [1] (17%). Surprisingly many (81%) have also rejected permission requests, and some have even uninstalled applications due to its EULA after installing it. The rejection number (81%) is much higher than found in previous research on the subject of permissions [1] (20%). This provides us with three clear indications that the general awareness and care for privacy issues in our principal inquiry user group is significant, and that the specific feedback we received in our questions below, on actor trust as well as privacy aspects, therefore is likely to be after carefully being considered by the respondents.

## 3.2 User attitudes

We investigated user attitudes toward sharing location and identity information about themselves, using a number of questions as this was key to our interests. Specifically, we asked:

- Do you tag friends in photos you publish online?
- Do you ask friends for permission to publish photographs of them online?
- Do you ask friends not to tag or publish photos of you online?

- Do you ask people you visit in their homes for permission to publish location information online?
- Do you add location information to photos you publish online?
- Do you tag location checkins with friends?
- Do you ask friends not to tag or publish checkins with you online?

Figure 6 shows that tagging photos with friend identities was more common that sharing location information, and that tagging friends, asking about permission to tag, and asking not to be tagged was equally common. Below, we break the responses we received down in more nuance as well as relate this to extant research where applicable.

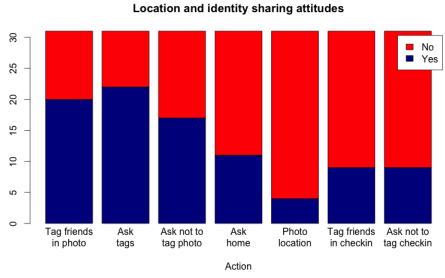



**Figure 6: Identity and location sharing attitudes**

*3.2.1 Proximity and Bluetooth.* In the survey we reminded the respondents that using Memorit they can set a reminder to trigger when in proximity (5-10m) of a friend, and that this means friends can do the same to the respondent. We then asked the respondents if they mind being discoverable by the phones of friends, friends of friends and anyone respectively. The result can be seen in Figure 7. respondents were asked to comment on their responses regarding proximity. Most users commented that as long as they could control who they were visible to, or could turn off visibility they were in general positive to being identifiable when in proximity to someone.

Related to proximity visibility, we asked about what respondents thought about lifelog cameras. The questions included if they want one, mind if other people nearby wear them, and would ask people not to use them? The results were inconclusive, with respondents divided and 16 positive vs. 15 negative. Given that lifelog cameras - while at one point heavily marketed - never became commonplace, we are not surprised at the results. Our desire when asking about these cameras, was to understand if the extreme case of increasing video photography using mobile phones or action cameras for selected situations, could carry more deeply rooted concerns to users or not. Asking this was also partly motivated by exploring if picture based analysis of context would be relevant to consider for future development of Memorit, but given the results the area remains for future research to nuance before making any design related changes.

Also, as Proximates use Bluetooth, and that Bluejacking has been expressed as a security concern, we wanted to find out if Bluejacking is an actual threat that has influenced the attitudes concerning proximity detection. We found that 18% of the respondents had received unsolicited connection attempts over Bluetooth. This is a much higher number than we expected. To our knowledge there has been no previous research that measured and reported number of attacks on users. We suspect Bluejacking is more common among young people and in schools as all who had received unsolicited Bluetooth connection attempts were born 1993 or later, which indicates they were likely students.

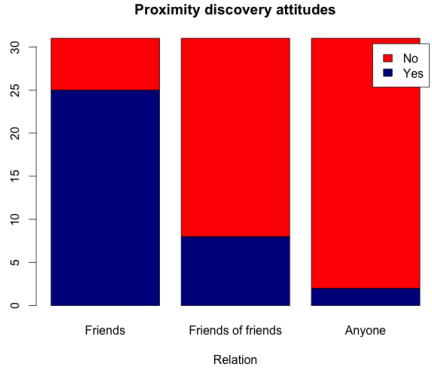



**Figure 7: Proximity discovery attitudes**

*3.2.2 Discoverable and checkin tagging.* Devising our questions, we hypothesized that users who ask to not be tagged in location check-ins by friends, would also not accept being proximity-discoverable by friends (and vice versa). The same question was then asked regarding friends of friends. The reasoning behind this hypothesis is that the type of information and the reach is similar and may thus hold similar response. The survey results, however, can neither confirm nor reject this without expanding the study beyond our control group. Our data indicates that users who asked friends to not tag them in check-ins, still mostly accepted being discoverable by both friends and friends of friends. Users who said nothing to their friends in regards to tagging mostly accepted being discoverable by friends, but not friends of friends (see table 1). A possible explanation for this is that users who ask friends to not tag them feel they are in control of the information about them being shared and thus comfortable with fiends of friends discovering them, while those who do not ask to tag feel less in control and thus less comfortable with being discoverable by anyone else than their friends.

*3.2.3 Severity of damage.* Proximity information can be used as a proxy for information about physical social interactions [3]. We therefore asked users how severe or worrying it would be if a

| | | Ask no to tag | |
|---|---|---|---|
| | | Yes | No |
| Discoverable | Yes | 8 | 17 |
| | No | 1 | 5 |

**Table 1: Discoverable by friends vs ask not to tag**

| | | Ask no to tag | |
|---|---|---|---|
| | | Yes | No |
| Discoverable | Yes | 7 | 1 |
| | No | 2 | 21 |

**Table 2: Discoverable by friends of friends vs ask not to tag**

third party (such as an individual person, company, or government agency) gained access to sensitive information about the location of the user? We also asked the same question in relation to who they were together with at some time, to understand if this impacted the answer. The response to these questions was identical for every respondent on both questions: 11% though it very severe, 38% thought it somewhat severe and 51% thought it not severe or worrying at all. This clearly indicates that who this third party is does not affect their stance. Instead, our respondents appear to be arguing that 'if one person can access the data, anyone potentially can'. Our data also indicates that neither location nor proximity is a deciding factor to our respondents, but rather that someone can access some form of personal information. What this data type is, does not carry particular weight to users.

## 4 SUMMARY OF FINDINGS

Given the low number of existing proximity based applications diffused among consumers today, we are not surprised that research into proximity based privacy concernes is somewhat lagging. To address this, we designed, developed and recruited users to a proximity based reminder application to give our respondents hands on experience with proximity services. 100 phones where handed out with this proximity reminder application pre-installed. After using the application on a daily basis, these users where queried through a text message based survey. Out of the specific answers to each question posed, we want to highlight four main insights. While all results carry weight, these four stand out as they go somewhat beyond our expectations as the study was designed.

First, we note that our respondents are clearly more aware of and care about privacy aspects in general than what previous studies indicate. This is an indication of relevance to our other findings, as it strengthens the validity of the response we have received there. As our general questions on privacy were only to position our respondents in relation to previous studies, we do not interpret our findings as signs of significant change in relation to such previous studies, but more research may be relevant to follow up if this was an anomaly in our respondents being particularly caring or if the trend is changing and EULA or similar privacy policies are becoming more relevant. Second, we see that respondents make decisions on trusting application developer based on the permissions they request rather than through reading EULAs. This is interesting in particular as our respondents apparently do also read the EULAs to a higher degree than in previous studies. Thirdly, that our respondents are willing to share identity, location and proximity information, as long as they are in control of who can access it. User control thus appears to play a particular role in trust building towards users. Fourthly, and in terms of severity if there is a privacy breach, there is no perceived difference in sensitivity between the different types of information. We hypothesize that this is due to differences in perceived control (or implicitly: trust), and that it is the breach in this control that is the driving factor in terms of privacy - not which type of information is accessed.

## ACKNOWLEDGMENTS

This work was partly funded by the two research centers: the Industrial Excellence Center Embedded Applications Software Engineering (EASE) (http://ease.cs.lth.se), and the Swedish Knowledge Foundation co-funded Internet of Things and People research profile (IoTaP) (http://iotap.mau.se/).

## REFERENCES

[1] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*. ACM, 3.

[2] Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Patrick Kelley, Madhu Prabaker, and Jinghai Rao. 2009. Understanding and capturing people's privacy policies in a mobile social networking application. *Personal and Ubiquitous Computing* 13, 6 (2009), 401–412.

[3] Vedran Sekara and Sune Lehmann. 2013. Application of network properties and signal strength to identify face-to-face links in an electronic dataset. (2013), 1–11. arXiv:arXiv:1401.5836v1

[4] Margaret Tan and Kathrine Sagala Aguilar. 2012. An investigation of students; perception of Bluetooth security. *Information Management; Computer Security* 20, 5 (2012), 364–381.

[5] Tony Vila, Rachel Greenstadt, and David Molnar. 2003. Why we can't be bothered to read privacy policies models of privacy economics as a lemons market. *Proceedings of the 5th international conference on Electronic commerce ICEC 03* (2003), 403–407. http://portal.acm.org/citation.cfm?doid=948005.948057

# Temporal Limits of Privacy in Human Behavior

**Vedran Sekara**[a,b,1,*], **Enys Mones**[a, b], **and Håkan Jonsson**[a,c,1,*]

[a]Sony Mobile Communications, SE-22188 Lund, Sweden; [b]Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark; [c]Faculty of Engineering (LTH), University of Lund, SE-22100 Lund, Sweden

**Large-scale collection of human behavioral data by companies raises serious privacy concerns. We show that behavior captured in the form of application usage data collected from smartphones is highly unique even in very large datasets encompassing millions of individuals. This makes behavior-based re-identification of users across datasets possible. We study 12 months of data from 3.5 million users and show that four apps are enough to uniquely re-identify 91.2% of users using a simple strategy based on public information. Furthermore, we show that there is seasonal variability in uniqueness and that application usage fingerprints drift over time at an average constant rate.**

privacy | computational social science | data mining | metadata

Tracking behavior is a fundamental part of the emerging big-data economy, allowing companies and organizations to segment, profile and understand their users in increasingly greater detail. Modeling context and interests of users has proven to have various advantages: products can be designed to better fit customers' needs; content can be adapted; and advertising can be made more relevant (1–6). Efficient user modeling requires the collection of large-scale datasets of human behavior, which has led to a growing proportion of human activities to be recorded and stored (7). Today, most of our interactions with computers are stored in a database, whether it is an e-mail, phone call, credit-card transaction, Facebook like, or online search, and the rate of information growth is expected to accelerate even further in the future (8). These rich digital traces can be compiled into detailed representations of human behavior and can revolutionize how we organize our societies, fight diseases, and perform research; however, they also raise serious privacy concerns (9–16). For example, Narayanan et al. demonstrated the feasibility of inferring political views of IMDb users through re-identification of movie ratings (17). Another infamous case is the hacking (and eventual erasure of personal data) of multiple accounts of a journalist, which was carried out by the attacker being able to connect two different databases (18).

The ubiquity and sensing capabilities of mobile phones together with our seemingly symbiotic relationship to them, renders these devices good tools for tracking and studying human behavior (19, 20). Mobile phones are ubiquitous and have permeated nearly every human society: in the year 2015 98.3% of the world's population had a mobile subscription (21). Mobile phones have transformed the way people access the internet as well: today the majority of traffic to web pages stems from mobile devices rather than from desktop computers (22), making advertisers target mobile phones to a higher degree. With the standard methods based on cookies for identifying customers not being used in smartphone apps, along with the rising usage of ad-blockers among users (23), advertisers and so-called *data brokers* are now targeting smartphone applications to replace the rich data cookies provided in the past. Advertisement identifiers are one such ID embedded in applications, but they do not allow data brokers to track users across multiple applications or devices, and they can even be reset by the user. Application usage behavior, however, cannot be cleared, and it is hard (and in many cases not feasible) to be changed or manipulated by users. This creates an economic incentive for global population tracking of application usage. This tracking is in conflict of users' perception of permissible usage of data (24). Also, in general, users are not knowledgeable enough about what data is collected about them to make an informed decision (25).

A majority of the online services people interact with on a daily basis collect personal information and sell the data to data brokers (third parties) (26). In a recent report released by the U.S. Federal Trade Commission, it was shown that data broker companies obtain vast amounts of personal data, which they further enrich with additional online and offline sources, and re-sell these improved datasets to the highest bidder, typically without the explicit consent or knowledge of the users (27). According to U.S. privacy laws, data is considered anonymous if it does not contain personally identifiable information (PII) such as name, home address, email address, phone number, social security number, or any other obvious identifier. As a result, it is legal for companies to share and sell anonymized versions of a dataset. However, as studies have shown, the mere absence of PII in a dataset does not necessarily guarantee anonymity due to the fact that it is relatively easy to compromise the privacy of individuals (11, 17, 28).

Human behavior, although imbued with routines, is inherently diverse. Previous work has shown that 99.4% of smartphone users have unique app usage patterns and established the viability of using apps as markers of human identity, similar in application to fingerprints in forensic science (29–31). It has further been demonstrated that the software infrastructure we use to access the Internet can be used to identify users (10). The digital breadcrumbs we leave online can be used to infer many aspects of our lives. It has been shown for example that age, gender, relationship status, education level, political beliefs, sexual orientation, religion, and even personality can be predicted from Facebook likes (32, 33), or based on the apps people use on their smartphones (34–36). Human mobility traces has been shown to be highly unique and research has further shown that 4 spatio-temporal points are sufficient to re-identify a majority of individuals (11).

This study demonstrates how easy it is to uniquely identify individuals from their smartphone usage patterns given only a handful of data points, and investigates the temporal patterns of uniqueness, revealing that humans are easier to identify

V.S. and H.J. conceived the study. V.S. and H.J designed measures and analyses. V.S. collected and curated the data. V.S. conducted the analysis. V.S., E.M., and H.J. wrote the manuscript. All authors interpreted and discussed the findings.

The authors have no conflict of interest.

[1]V.S. and H.J. lead the research and contributed equally to this work.

[*]To whom correspondence should be addressed: hajons@gmail.com, vedransekara@gmail.com.

during certain periods of the year. We define identification as matching a behavior pattern against an (anonymous) quasi-identifier consisting of a similar pattern. In the dataset we use, no further information can be gained about the user beyond matching two patterns. However, in a real world scenario, an attacker could use this method for connecting two datasets to learn new information about the re-identified user, e.g. email address, age, or gender, depending on the data available to the attacker. Our study focuses on applications (apps) — small software programs which users can download to their smartphones, and which provide a near unlimited range of functions, from simple functions such as flashlights or calculators to more advanced—artificial intelligence like—functions. Each new phone comes with a set of apps pre-loaded by the manufacturer, but a user is free to customize their device to suit their specific needs, as such users have access to millions of apps on app stores such as *Google Play* (approx. 2.8 million apps) (37).

## Results

**Uniqueness of human behavior.** To evaluate the likelihood of identifying individuals within smartphone usage data we use a dataset that spans 12 months (Feb. 1st 2016 to Jan. 31st 2017) and encompasses 3.5 million people using in total 1.1 million unique apps. We have chosen to disregard phone vendor specific apps, such as alarm clock apps, built-in phone dialer apps, etc. and only focus on apps that are downloadable from Google Play. From this we form app *fingerprints* for each user, i.e. a binary vector containing information about which apps the user has used for every month. We only consider apps actually used by a user in a month, not apps that were installed but never used. Figure 1 illustrates the typical patterns of app usage, with individuals continuously changing their app-fingerprint over the course of a year by trying out new apps and ceasing to use others. As such, app-fingerprints slowly drift over time, with the average rate of change being roughly constant between consecutive months (Figure S1). In combination with fingerprints drifting, the number of apps people use on their smartphones is constant over time as well, suggesting that humans have a limited capacity for interacting, navigating, and managing the plethora of services and social networks offered by smartphones (Figure S2). This limiting effect has been observed in other aspects of life such as interactions among people (38) or geo-spatial exploration (39).

The risk of re-identifying individuals is estimated by means of unicity (11, 14). Here, re-identification corresponds to successful assignment of an app-fingerprint to a single unique user in our dataset. This does not entail that we can directly get the *real* identity of a person, such as name, address, e-mail, social security number, etc. This, however, would become possible if this knowledge is cross-referenced with other data sources, which there unfortunately has been countless examples of (17, 40–43). Given an individual's app-fingerprint, unicity quantifies the number of apps needed to uniquely re-identify that person; the fewer apps we need the more unique a person is and vice versa. Given a dataset of app-fingerprints and set of apps $i$, $j$ and $k$, a user $u$ is uniquely identifiable if that user, and only that user, in the dataset has used apps $i$, $j$ and $k$, i.e. matching the fingerprint of user $u$. In our dataset we evaluate uniqueness as the percentage of users we can re-identify using $n$ number of apps.

To attack the dataset without any prior knowledge of the system itself, the most realistic strategy is to pick apps at random. Figure 2A shows the efficiency of this type of random sampling of apps, with 21.8% of users being re-identified from using 4 apps. Although this value means only 1 of every 5 individual can be re-identified, it is surprisingly high given that we only use binary features (that is, has the user used the app or not) and have no information regarding *when* an app was used or for *how long*—features which would only make fingerprints more unique. In case of a real attack, however, the above results might give the general public a false sense of security as it is possible to use free, publicly available information to formulate an attack strategy that greatly outperforms the random strategy.

The popularity of apps follows a heavy-tailed distribution (44) (and see Figure S3); a few apps are used by millions or even billions of individuals, while an overwhelming majority of apps only have a couple of users. All this information is available on *Google Play* from where it is possible to retrieve by automatic means, or it can be purchased from vendors such as AppMonsta. Because this information is so easily attainable, we formulate a strategy that takes the user base of apps (popularity of apps) into account, starting with the least used apps: the *popularity strategy*. Rather than using the popularity in terms of downloads on Google Play, we use the popularity counted as the number of users that use an app in our dataset (see Methods for details). A real-world re-identification attack strategy could use the Google Play download numbers for each app to reduce the amount of computation required. Figure 2B shows that just using 2 apps with the popularity strategy greatly outperforms the random strategy, and using 4 apps, we are able to re-identify 91.2% of users.

**Seasonal variability of anonymity.** Human lives, routines and behaviors evolve over time (39, 45, 46), and therefore individual app-fingerprints might become harder (or easier) to identify. To quantify the seasonal variability of uniqueness, we construct monthly fingerprints for all individuals and evaluate anonymity using the unicity framework. Figure 3 shows the fraction of individuals that are re-identifiable per month, and reveals an increased fraction of identifications for June, July, and August—months which are typically considered vacation months. The increase in uniqueness is independent of how we select apps (random, or by popularity). In fact, during these three months the process of identifying individuals from randomly selected apps is respectively 14.8% and 18.4% more effective when using 5 and 10 apps. For the popularity scheme, we note 6.8% and 8.0% higher rates of identifications when using 5 and 10 apps. The increase in identifiability stems from a combination of related behavioral changes (Figure S4). Apps related to categories such as travel, weather, sports, and health & fitness gain popularity during the summer months (June, July, August), related to people traveling and downloading apps that help them navigate new cities, using fitness apps to motivate them to exercise more, and using apps that enable them to follow global sports events such as the 2016 UEFA European Championship in football (soccer). Simultaneously, apps related to categories such as education and business become less popular. This suggests an interplay between our physical behavior and our app-fingerprint, indicating that when we change our geo-spatial routines by traveling and

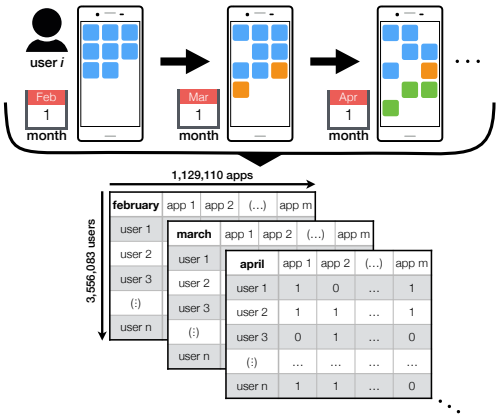



**Fig. 1.** Smartphone usage patterns change over time, with users continuously changing which apps they use. This study is based on smartphone app-fingerprints of 3,556,083 individuals. For each month between February 2016 and January 2017, we retrieve the list of apps a person has used during the period ($n_{\text{month}} = 23$ apps per person per month on average, or $n_{\text{year}} = 76$ apps on average during the full 12-month period). App-fingerprints are represented as a sparse *user* $\times$ *app* $\times$ *month* tensor, with 1 indicating a person has used an app during a specific month, 0 otherwise. To look at longer time-windows, we aggregate entries according to a maximum value heuristic and retain entries if they are greater than zero.

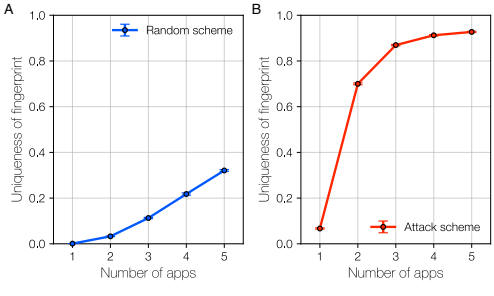



**Fig. 2.** Uniqueness of smartphone app-fingerprints given $n$ number of apps. (A) Selecting apps at random is not an efficient way of identifying individuals and achieves a modest re-identification rate of $21.8\%$ when using 4 apps. (B) Using freely available outside information from Google Play to attack the problem yields significantly higher rates of re-identifications, $91.2\%$ when using 4 apps. Error bars denote one standard deviation. App-fingerprints are constructed from the full 12 months of data, and $99.7\%$ of individuals within our dataset have a unique fingerprint.

exploring new places, we also change our app usage. This change in phone behavior makes our app-fingerprints more unique and easier to identify.

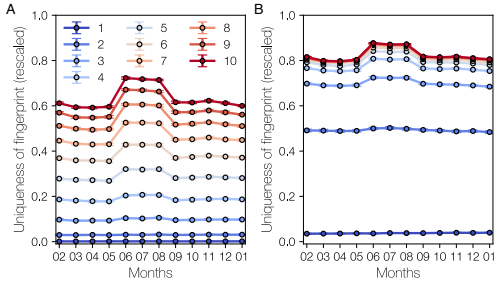



**Fig. 3.** Seasonal variations of re-identifiable app-fingerprints over 12 months. The fraction of individuals which we can re-identify by using $n$ apps (1-10) changes from month to month, revealing that uniqueness has a temporal component, and that people are more unique during summer. This is independent of whether apps are selected using: (A) a random heuristic or (B) an attack scheme. Compared to Figure 2, the fraction of re-identified individuals per month is lower because we have segmented behavior into monthly fingerprints as compared to constructing fingerprints from 12 months of data. Uniqueness is rescaled according to the set size of apps present within each month (see Figure S5).

**Hiding in the crowd.** Our dataset is limited to 3.5 million users, similar in size to a small country, but how will uniqueness change as more users are added (increased sample-size)? Will it become possible to hide in the crowd? More precisely, how does the population size affect the extent to which a specific app-fingerprint remains unique. That is, as more and more users are added to our sample, does the likelihood to observe multiple individuals with identical fingerprints also increase? This corresponds to an inverse k-anonymity problem (28), where one needs to estimate the number of users that should be added in order to increase the overall anonymity of the dataset. (Bearing in mind that overall anonymity is not a good measure for the sensitivity of individual traces.) To understand the effect of sample-size on unicity, we first slice our dataset into smaller subsamples and use it to estimate the uniqueness for sample sizes ranging from 100,000 to 3.5 million individuals. Figure 4A reveals that sample size has a large effect on the re-identification rate when selecting apps using a random heuristic. Considering $n_{\text{apps}} = 5$, the average re-identification rate decreases from $45.89\%$ for a sample size of 1 million individuals to $37.33\%$ for 2 million individuals and $32.09\%$ for the full sample of 3.5 million people. The attack scheme is considerably less affected (Figure 4B). For $n_{\text{apps}} = 5$ we find that the re-identification rates are respectively $96.60\%$, $94.23\%$ and $92.72\%$ for sample sizes of 1, 2 and 3.5 million individuals. As such, increasing the sample size by $250\%$ (from 1 to 3.5 million individuals) only reduces uniqueness by approximately 4 percent-points.

In order to estimate uniqueness for sample sizes larger than the study population we extrapolate results from Figure 4B for $n_{\text{apps}} = 5$. We express uniqueness of fingerprints using multiple functional forms including: power-laws ($\sim x^{\gamma}$), exponentials ($\sim \exp(\gamma x)$), stretched exponentials ($\sim \exp(x^{\gamma})$), and linear functions ($\sim x$), where $x$ denotes the sample size and $\gamma$ is a

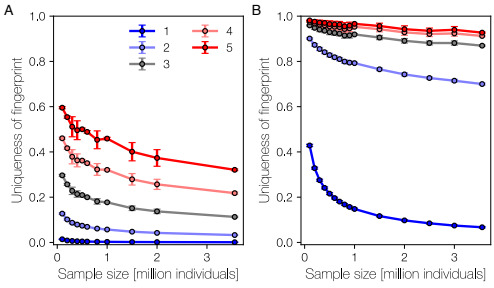



**Fig. 4.** Identifying fingerprints across data-samples with varying population sizes. Fingerprints are constructed from 12 months of data. The uniqueness of individual fingerprints is reduced (lower re-identification rates) as we increase the sample-size independently of whether apps are selected: (A) randomly or (B) according to the attack heuristic. The magnitude of the change, however, varies greatly different between the two heuristics. Results show in both panels are calculated from multiple realizations of the data (see Materials and Methods section).

scaling factor. The stretched exponential and power-law show the highest agreement with the data (Figure S6), and roughly suggest that 5 apps are enough to re-identify 75%–80% of individuals for 10 times larger samples (35 million individuals). Although the applied analysis displays high uncertainty with regards to extrapolations, it illustrates the observation that increasing the population size does not help us in hiding in the crowd (that is, uniqueness is not a characteristic of small sample sizes).

## Discussion

Phone behavior is different from credit card traces and mobile phone mobility data in that the ease with which data can be collected, and any Android app can request permission to access your app history. We reviewed apps with more than 100,000 downloads which request the 'retrieve running apps' permission on Android, and that are free (no price or in app purchases). Out of these 40 apps 31 contain ads. There are 15 apps that belong to the Personalization or Tools category, mostly anti-virus or launcher apps, which may need the permission to provide their features. For the other 25 apps, we found no features in the app that would motivate requesting this permission. Some of these apps are from major phone vendors whose privacy policy says they may share data with third parties.

The economic incentives, the easy and global scale of collecting and trading this data without users' knowledge creates some serious concerns, especially since this practice is in violation of users' expectations or knowledge (24, 25). The EU General Data Protection Regulation (GDPR) may be a first step towards addressing these concerns through regulation, since it does mention unicity (47) and applies globally to data about any EU citizen. Our conclusion from this study is that application usage data should be considered personal information, since it is a unique fingerprint.

This study was performed using app usage data collected from Android phones from a single vendor only. As phone vendor specific apps were disregarded in the analysis, we expect the results to generalize across all Android devices. Further, we have no reason to believe that app usage behaviour and uniqueness is fundamentally different for individuals using iOS devices compared to Android users.

## Materials and Methods

**The dataset.** We use a dataset that spans 12 months, from Feb. 1st 2016 to Feb. 1st 2017, and contains monthly app-fingerprints for 3,556,083 individuals with pseudonymized app and user identifiers. Each fingerprint is a binary vector composed of the apps a person has used during a month. We do not consider apps that are installed but unused. We further disregard phone vendor specific apps such as: alarm clock, phone dialer, settings etc. and only focus on apps that are downloadable from Google Play. This removes vendor bias, and makes re-identification harder. The users are selected from major markets in the Americas, Europe and Asia. Thus, the impact of regional variations on uniqueness due to local applications is smaller than if we had sampled users from anywhere in the world. In total, the number of unique apps in the dataset is 1,129,110, and each individual in the dataset uses at least 3 apps per month. Data collection is approved by the Sony Mobile Logging Board and written consent in electronic form has been obtained for all study participants according to the Sony Mobile Application Terms of Service and the Sony Mobile Privacy Policy. Raw data cannot be shared publicly on the web, but we offer the possibility to reproduce our results starting from raw records by spending a research visit at Sony Mobile Communications.

**Estimating uniqueness.** To estimate the uniqueness of app-fingerprints, we apply the unicity framework (11) on $k$ samples of 10,000 randomly selected individuals. For each individual we select $n$ apps (without replacement) from the person's app-fingerprint. With the popularity based attack, apps with low user base are selected to increase the uniqueness of the app usage pattern. The person is then said to be unique if they are the only individual in the dataset whose app-fingerprint contains those apps. In cases where $n$ is larger the the total length of a person's app-fingerprint we instead select $\min(n, |\text{fingerprint}|)$ number of apps. Uniqueness for a sample $k_i$ is then estimated as the fraction of the users that have unique traces. Overall uniqueness is the average of the $s$ samples, and error-bars are given by the standard deviation. We use $s = 20$.

**Subsampling the dataset.** To quantify the relation between sample size and uniqueness, we subsample the dataset by selecting a fraction of the original dataset. For each sample $s_i$ we estimate uniqueness using the above methodology. To account for selection bias we estimate uniqueness as the average of multiple realizations of a sample size. We use 20 realizations for sample sizes between 100,000 - 500,000, 10 realizations for samples between 600,000 - 900,000, and 5 realizations for sample sizes above 1,000,000 individuals.

**ACKNOWLEDGMENTS.** V.S. and H.J would like to thank Sune Lehmann for useful discussions and feedback.

1. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. *SIGMOD Rec.* 22(2):207–216.
2. Bell RM, Koren Y (2007) Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter* 9(2):75–79.
3. Chen Y, Pavlov D, Canny JF (2009) Large-scale behavioral targeting in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. (ACM), pp. 209–218.
4. Mislove A, Viswanath B, Gummadi KP, Druschel P (2010) You are who you know: inferring user profiles in online social networks in *Proceedings of the third ACM international conference on Web search and data mining*. (ACM), pp. 251–260.
5. Dodds PS, Danforth CM (2010) Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of happiness studies* 11(4):441–456.
6. Mislove A, Lehmann S, Ahn YY, Onnela JP, Rosenquist JN (2011) Understanding the demographics of twitter users. *ICWSM* 11:5th.
7. Conte R, et al. (2012) Manifesto of computational social science. *European Physical Journal-Special Topics* 214:p–325.
8. Lazer D, et al. (2009) Computational social science. *Science* 323(5915):721–723.
9. Blumberg AJ, Eckersley P (2009) On locational privacy, and how to avoid losing it forever. *Electronic Frontier Foundation* 10(11).
10. Eckersley P (2010) How unique is your web browser? in *International Symposium on Privacy Enhancing Technologies Symposium*. (Springer), pp. 1–18.
11. De Montjoye YA, Hidalgo CA, Verleysen M, Blondel VD (2013) Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* 3:1376.
12. Hannak A, et al. (2013) Measuring personalization of web search in *Proceedings of the 22nd international conference on World Wide Web*. (ACM), pp. 527–538.
13. Greenwood D, Stopczynski A, Sweatt B, Hardjono T, Pentland P (2014) The new deal on data: A framework for institutional controls. *Privacy, big data, and the pubic good* pp. 192–210.
14. De Montjoye YA, Radaelli L, Singh VK, , et al. (2015) Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science* 347(6221):536–539.
15. Sapiezynski P, Stopczynski A, Gatej R, Lehmann S (2015) Tracking human mobility using wifi signals. *PloS one* 10(7):e0130824.
16. Mayer J, Mutchler P, Mitchell JC (2016) Evaluating the privacy properties of telephone metadata. *Proceedings of the National Academy of Sciences* p. 201508081.
17. Narayanan A, Shmatikov V (2008) Robust de-anonymization of large sparse datasets in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. (IEEE), pp. 111–125.
18. Honan M (2012) How Apple and Amazon security flaws led to my epic hacking (https://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/).
19. Eagle N, Pentland AS (2006) Reality mining: sensing complex social systems. *Personal and ubiquitous computing* 10(4):255–268.
20. Stopczynski A, et al. (2014) Measuring large-scale social networks with high resolution. *PloS one* 9(4):e95978.
21. Union IT (2016) International Telecommunication Union, World Telecommunication/ICT Development report and database.
22. Enge E (2017) Mobile vs desktop usage: Mobile grows but desktop still a big player. https://www.stonetemple.com/mobile-vs-desktop-usage-mobile-grows-but-desktop-still-a-big-player/.
23. PageFair (2017) The state of the blocked web - 2017 global adblock report. https://pagefair.com/blog/2017/adblockreport/.
24. Martin K (2018) The penalty for privacy violations: How privacy violations impact trust online. *Journal of Business Research* 82:103–116.
25. Posner RA (1981) The economics of privacy. *The American economic review* 71(2):405–409.
26. Anthes G (2015) Data brokers are watching you. *Communications of the ACM* 58(1):28–30.
27. Ramirez E, Brill J, Ohlhausen MK, Wright JD, McSweeny T (2014) Data brokers–a call for transparency and accountability. *Federal Trade Commission, Tech. Rep.* www.ftc.gov/system/files/documents/reports/data-brokers-call-transparency-accountability-report-federal-trade-commission-may-2014/140527databrokerreport.pdf.
28. Sweeney L (2002) k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05):557–570.
29. Falaki H, et al. (2010) Diversity in smartphone usage in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. (ACM), pp. 179–194.
30. Welke P, Andone I, Blaszkiewicz K, Markowetz A (2016) Differentiating smartphone users by app usage. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* pp. 519–523.
31. Achara JP, Acs G, Castelluccia C (2015) On the unicity of smartphone applications in *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society*. (ACM), pp. 27–36.
32. Kosinski M, Stillwell D, Graepel T (2013) Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences* 110(15):5802–5805.
33. Youyou W, Kosinski M, Stillwell D (2015) Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences* 112(4):1036–1040.
34. Chittaranjan G, Blom J, Gatica-Perez D (2013) Mining large-scale smartphone data for personality studies. *Personal and Ubiquitous Computing* 17(3):433–450.
35. Seneviratne S, Seneviratne A, Mohapatra P, Mahanti A (2014) Predicting user traits from a snapshot of apps installed on a smartphone. *ACM SIGMOBILE Mobile Computing and Communications Review* 18(2):1–8.
36. Malmi E, Weber I (2016) You are what apps you use: Demographic prediction based on user's apps. *Tenth International AAAI Conference on Web and Social Media*.
37. AppBrain (2017) Google Play statistics (https://www.appbrain.com/stats). Accessed: 2017-04-18.
38. Dunbar RI (1992) Neocortex size as a constraint on group size in primates. *Journal of human evolution* 22(6):469–493.
39. Alessandretti L, Sapiezynski P, Lehmann S, Baronchelli A (2016) Evidence for a conserved quantity in human mobility. *arXiv preprint arXiv:1609.03526*.
40. Barbaro M, Zeller T, Hansell S (2006) A face is exposed for aol searcher no. 4417749. *New York Times* 9(2008):8For.
41. Barth-Jones DC (2012) The're-identification'of Governor William Weld's medical information: a critical re-examination of health data identification risks and privacy protections, then and now. *Available at SSRN: https://ssrn.com/abstract=2076397*.
42. Sweeney L, Abu A, Winn J (2013) Identifying participants in the personal genome project by name. *Available at SSRN: https://ssrn.com/abstract=2257732*.
43. Tockar A (2014) Riding with the stars: Passenger privacy in the nyc taxicab dataset. *Neustar Research, September* 15.
44. Olmstead K, Atkinson M (2016) Apps permissions in the Google Play store. *Pew Research Center*.
45. Kossinets G, Watts DJ (2006) Empirical analysis of an evolving social network. *Science* 311(5757):88–90.
46. Sekara V, Stopczynski A, Lehmann S (2016) Fundamental structures of dynamic social networks. *Proceedings of the national academy of sciences* 113(36):9977–9982.
47. (2016) Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union* L119:1–88.

# Temporal Limits of Privacy in Human Behavior
# *Supplementary Information*

Vedran Sekara, Enys Mones & Håkan Jonsson

May 29, 2018

## S1 The dataset

We use a dataset that spans 12 months, from Feb. 1st 2016 to Feb. 1st 2017, and contains monthly app-fingerprints for 3,556,083 individuals. Each fingerprint is a binary vector composed of the apps a person has used during a month. We do not consider apps that are installed but unused.

We further disregard phone vendor specific apps such as: alarm clock, phone dialer, settings etc. and only focus on apps that are downloadable from Google Play. This removes vendor bias, and makes re-identification harder. The users are selected from major markets in the Americas, Europe and Asia. Thus, the impact of regional variations on uniqueness due to local applications is smaller than if we had sampled users from anywhere in the world.

In total, the number of unique apps in the dataset is 1,129,110, and each individual in the dataset uses at least 3 apps per month.

The data was collected using a pre-loaded app recommender app on Xperia phones. Data collection is approved by the Sony Mobile Logging Board and written consent in electronic form has been obtained for all study participants according to the Sony Mobile Application Terms of Service and the Sony Mobile Privacy Policy.

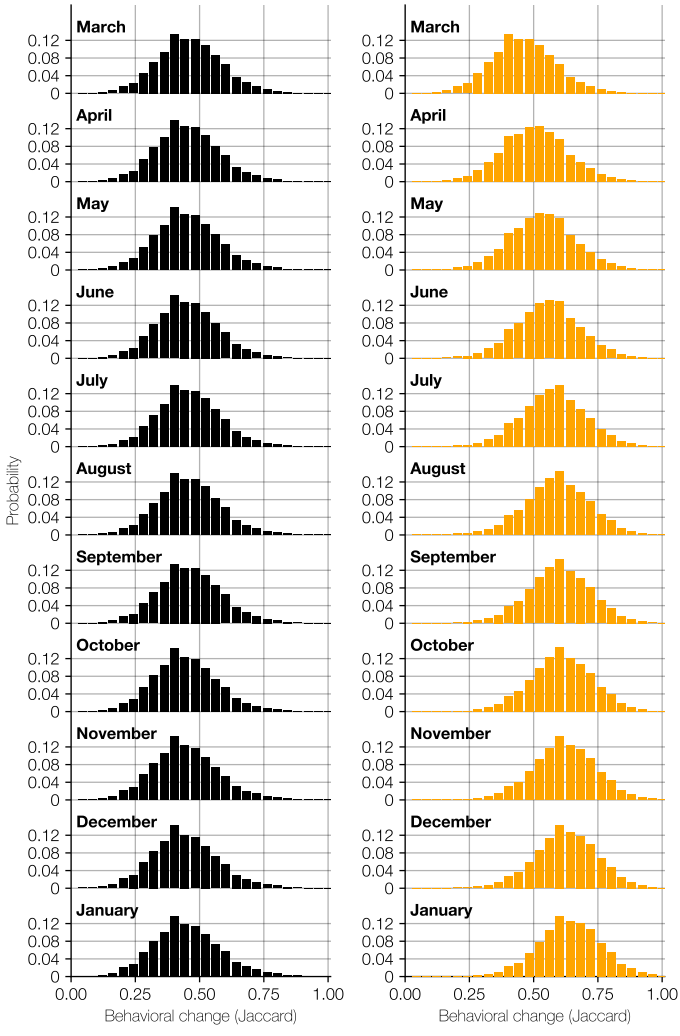



**Figure S1:** *Distributions show the change in app fingerprint over time. The change is measured as Jaccard distance between a users fingerprint in one month and the next. Left, change between consecutive months, e.g. February and March (denoted March), March and April (denoted April), etc. Right, Difference between fingerprint in February 2016 compared to other months, indicating a drift over time.*

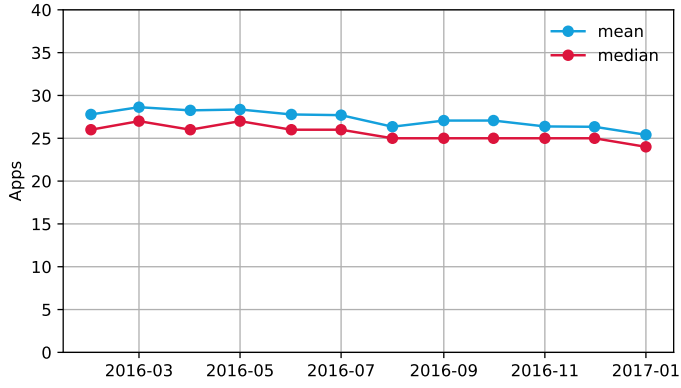



**Figure S2:** *Average number of apps per user per month. The median is also plotted for comparison.*

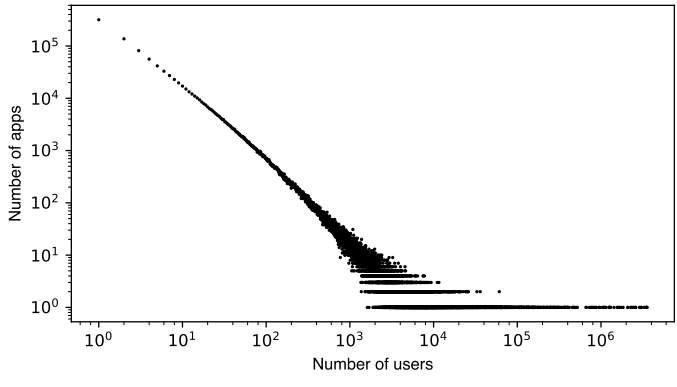



**Figure S3:** *Distribution of populatity of apps, i.e. the number of individuals using an app. Estimated across the entire dataset. Distribution clearly displays a long-tail.*

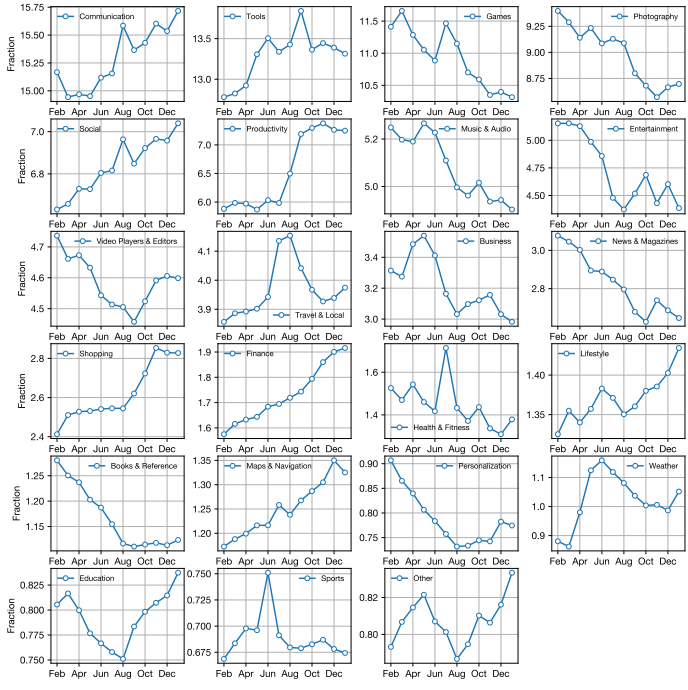



**Figure S4:** *Fraction of apps per category. Apps are divided into popular Google play categories and figure shows the fraction of app that belong to each category over time.*

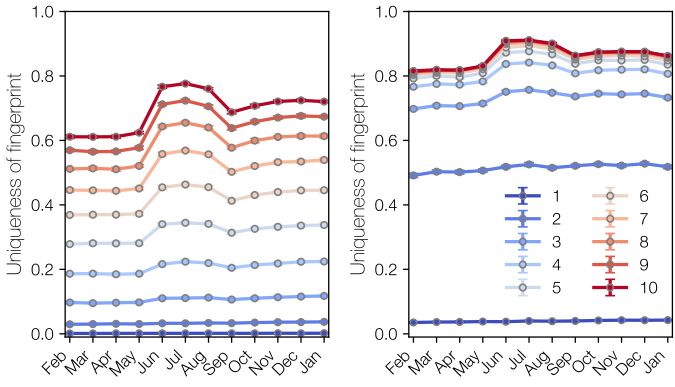



**Figure S5:** *Seasonable variations of uniqueness over time for the random scheme (left) and the popularity heuristic (right). Curves are rescaled according to* $\widetilde{u}(t) = \frac{u(t)}{|A|_t/|A|_{t=0}}$*, where* $u(t)$ *is the uniqueness at month* $t$*, and* $|A|_t$ *is the number of apps at time* $t$*. With* $t = 0$ *denoting the first month of the dataset, February 2016.*

## S2 Extrapolation to larger population

| Function | Pseudo $R^2$ | $a$ | $b$ | $\gamma$ |
|---|---|---|---|---|
| $ax^\gamma + b$ | 0.939 | -0.031 | 0.989 | 0.504 |
| $a\exp(x^\gamma) + b$ | 0.940 | -0.022 | 1.017 | 0.261 |
| $a\exp(\gamma x) + b$ | 0.931 | 0.066 | 0.914 | -0.388 |
| $ax + b$ | 0.908 | -0.014 | 0.975 | - |

**Table S1:** *Regression values.*

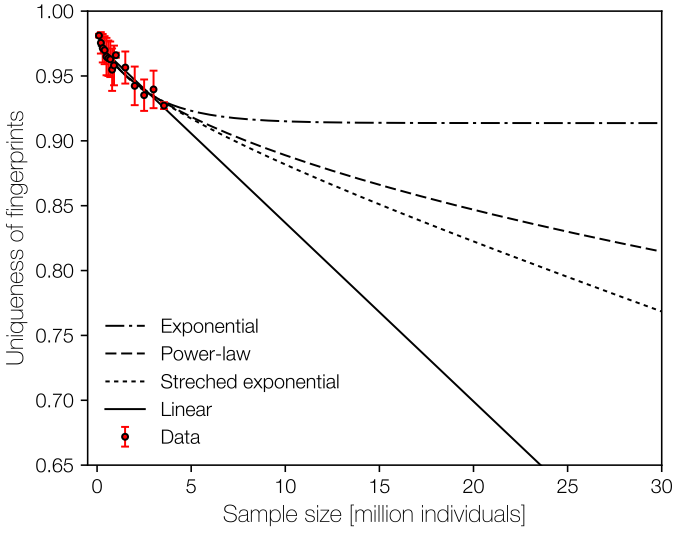



**Figure S6:** *Extrapolated uniqueness. Fit of different functional forms (see Table S1) to the uniqueness curve for $n_{apps} = 5$ when selecting apps using the popularity heuristic. Closes agreement with data is achieved by the stretched exponential and power law functional forms.*

# Möjligheter och faror med att brygga den digitala och den fysiska världen

Håkan Jonsson

## Teknikfrustration

Vår relation till maskiner karakteriseras ofta av frustration. Maskiner ska ju göra livet lättare genom att förenkla och hjälpa oss utföra uppgifter. De förstår oss dock inte på samma sätt som människor gör:  Vi tvingas använda gränssnitt som inte är optimala för varken maskinerna eller oss. För oss har en knapptryckning en betydelse och mening, och görs i en kontext av att vi vill utföra en uppgift. För maskinen är knappen bara en sensor som initierar en kausal kedja av händelser som resulterar i något slags effekt. Maskinerna missförstår oss inte heller på samma sätt som människor gör: När knapptryckningen inte resulterar i förväntad effekt så skulle de flesta av oss vilja förklara vår avsikt så att maskinen kan ändra sitt beteende nästa gång vi trycker; men maskinerna lyssnar sällan.

Med det stora genomslaget av smartphones i våra liv har potentialen för frustration växt, med denna maskins begränsade storlek och möjligheter för interaktion. Stora ansträngningar görs inom design av smartphones och dess applikationer samt inom forskningsfältet människa-dator-interaktion för att hantera detta problem. Ett sätt som visat sig fruktbart är att använda smartphonens många sensorer och möjligheter till datainsamling för att försöka härleda användarens *kontext* eller situation, och använda denna för att anpassa interaktionen. Det är nu därför vanligt för applikationer att använda vår identitet, tid, plats och aktivitet från dessa sensorer.
Till exempel kan applikationen Google Now visa en användare (*identitet*) som lämnar (*aktivitet*) kontoret (*plats*) när nästa buss åker (*tid*) hem.

## Social kontext

Dock finns det en typ av kontextinformation som är viktig i vårt dagliga liv, men som smartphones ännu inte kan använda sig av, nämligen *social kontext*. Vår sociala kontext utgörs av de människor vi interagerar med i vår nuvarande kontext, samt de relationer vi har till dem.

Detta orsakar frustration när vi vill använda vår smartphone till uppgifter som kräver denna sociala kontextinformation. Ett exempel på detta är när vi försöker dela kontaktinformation med någon i ett möte med hjälp av smartphonen. Om vi använder Bluetooth måste vi lista ut vilken telefon som hör till vilken person i rummet. Om vi använder LinkedIn så måste vi söka efter personen i applikationen. Det finns ingen koppling mellan den fysiska och sociala upplevelsen vi har och den digitala värld som vi vill dela information från. Båda dessa är exempel på fall då vi tvingas till interaktion som inte hade behövts om smartphonen hade kunnat förstå vilka personer som ingår i min sociala interaktion

och min relation till dem. Istället hade smartphonen detekterat vilka personer jag har ett möte med och frågat om jag vill skicka information till dem eller lägga till dem som kontakter.

I min forskning har jag utvecklat och utvärderat teknologier och metoder för att samla in och använda social kontextinformation och dess konsekvenser för integritet. Jag har också studerat vilka nya applikationer som möjliggörs och använt dessa för att genomföra användarstudier.

## Den sociala bryggan

För att koppla ihop den fysiska och den digitala social världen, dvs personerna vi träffar fysiskt och de vi träffar på t.ex. Facebook, har jag byggt en mjukvarukomponent som kan användas av smartphoneappar. Den använder Bluetooth, en radioteknik som kan detektera andra telefoner inom cirka 5m för att låta andra telefoner veta att den är nära samt för att detektera andra telefoner som är nära. För att koppla detekterad telefon till en användare registreras telefonens Bluetoothidentitet tillsammans med Facebookidentitet. Denna koppling gör det möjligt för applikationer att detektera vilka andra Facebookanvändare som är nära.

## Applikationer

Varför vill man då göra detta? Jag har genomfört en fältstudie där jag utvecklat ett flertal applikationer som använder bryggan för att undersöka nyttan. En av dessa är en smartphoneapp kallad Memorit (BIld 1) som gör det möjligt att skapa påminnelser för olika typer av händelser, t.ex. datum och tid, plats samt när man träffar en specifik person. Det kan vara användbart för få en påminnelse att betala tillbaka lunchpengar eller lämna en nyckel nästa.
Andra exempel är en fotoram som visar foton som innehåller de som är framför fotoramen, och en mötesapplikation som håller reda på vilka av de kallade som är närvarande (Bild 2) .

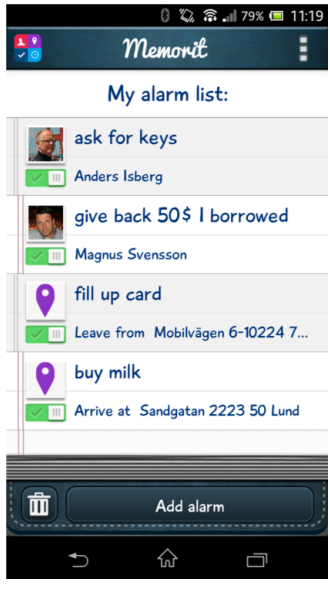


Bild 1

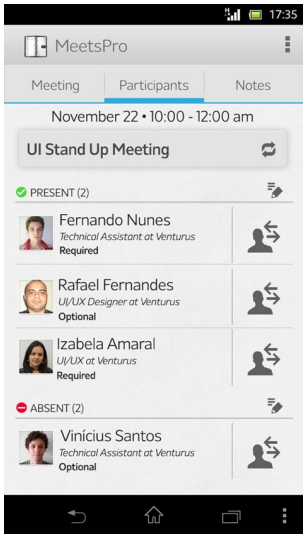


Bild 2

## Social agens

Dessa applikationer kan tyckas enkla men är exempel på en fundamental egenskap som hittills saknats i vår teknologi: Social agens. Med detta menar jag prylar och applikationer som kan representera oss som sociala individer, både vad gäller att känna igen andra individer och agera socialt å våra vägnar. I min forskning har jag dock visat att applikationer som bygger på social kontext kan vara svåra att sälja och förklara, då kontextinformation är så rik att vi har svårt att sätta oss in i andra kontext än det vi befinner oss i just nu. Detta har konsekvenser för applikationsutvecklare i ljuset av EUs nya datalagstiftning, som kräver att användaren förstår varför data samlas in. Vidare så kräver social agens att vi ger applikationer autonomi. Dock finns en applikationsspecifik kostnad för att begå misstag när denna autonomi utnyttjas, som applikationsutvecklare måste ta hänsyn till. Slutligen behöver utvecklare av produkter som är kapabla till social agens designa dem så att vi får ett s.k. införlivat (embodied) relation till dem. Detta betyder att de ska kännas som en förlängning av oss själva snarare än en separat pryl. Exempel på sådana produkter är t.ex. Kläder. Mobiltelefonen är dock en separat pryl som vi måste interagera med genom ett frustrerande gränssnitt.

## Integritet

Vad betyder då detta för vår integritet med avseende på datainsamling? Med smartphones har användare över tid vant sig och i större grad accepterat att applikationer använder mer och mer data och sensorer, t.ex. platsinformation. Dock har det vara oklart i vilken grad sensorinformation om vilka vänner som finns i omedelbar fysisk närhet uppfattas som känslig av användare. I min forskning har jag visat att det inte är någon skillnad på platsinformation och närhetsinformation i detta avseende, så länge användare har möjlighet att kontrollera vem som har tillgång till informationen. Dock finns ett annat problem: Insamlad information om användarkontext, t.ex. vilka applikationer man använt, samlas in och säljs av många applikationsutvecklare för riktad annonsering. Detta görs globalt och i extremt stor skala. Denna information har hittills setts som icke personlig data i juridisk mening. I en kvantitativ studie har jag dock visat att denna information utgör ett fingeravtryck som kan identifiera användare och därför ska ses som personlig. Även detta har konsekvenser för applikationsutvecklare i ljuset av GDPR och avslöjandet rörande Facebook och Cambridge Analytica.

The widespread adoption of smartphones with advanced sensing, computing and data transfer capabilities has made scientific studies of human social behavior possible at a previously unprecedented scale. It has also allowed context-awareness to become a natural feature in many applications using features such as activity recognition and location information.
However, one of the most important aspects of context remains largely untapped at scale, i.e. social interactions and social context. Social interaction sensing has been explored using smartphones and specialized hardware for research purposes within computational social science and ubiquitous computing, but several obstacles remain to make it usable in practice by applications at industrial scale.
In this thesis, I explore methods of physical proximity sensing and extraction of social context information from user-generated data for the purpose of context-aware applications. Furthermore, I explore the application space made possible through these methods, especially in the class of use cases that are characterized by *embodied social agency*, through field studies and a case study.
A major concern when collecting context information is the impact on user privacy. I have performed a user study in which I have surveyed the user attitudes towards the privacy implications of proximity sensing. Finally, I present results from quantitatively estimating the sensitivity of a simple type of context information, i.e. application usage, in terms of risk of user re-identification.

Department of Computer Science,
Lund University,
Box 118, SE-221 00 Lund, Sweden

ISBN 978-91-7753-689-5 (print)
ISBN 978-91-7753-690-1 (pdf)
Doctoral Thesis 1, 2018