Dissertation

An algorithmic approach to increase the context prediction accuracy by utilizing multiple context sources

zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften (Dr.-Ing.)

im Fachbereich Elektrotechnik/Informatik der

Universität Kassel

Immanuel König

Gutachter: 1. Prof. Dr.-Ing. Klaus David

2. Prof. Dr. Kristof Van Laerhoven

Datum der Einreichung: Kassel, November 2016

Datum der Disputation: Kassel, 21. Dezember 2016

Abstract

Today we are surrounded by technical systems and devices which aim to ease our live and support us in our daily activities. We use Smartphones to access information everywhere, intelligent home automation systems to gain security and comfort, and car assistance systems to drive more securely and comfortable. However, we often expect such systems to consider upcoming events: The Smartphone is expected to display the train delay before we begin our journey, the intelligent home automation is expected to air condition our home before we arrive and the car is expected to display the traffic jam before we get stuck in it. To achieve this, the systems monitor our behavior with sensors, process the collected sensor data in context sources and derive context values from it. Afterwards a context prediction algorithm predicts future upcoming context values, often only based on a single context source. While we expect the systems to predict as accurate as possible, there exists an upper boundary of how accurate a prediction algorithm can predict. This upper boundary can be calculated based on the context values. To increase the upper boundary and therefore to allow more accurate predictions, we propose the use of multiple context sources for context prediction. In this thesis

- an approach to combine multiple context sources for prediction is given,
- the gain of prediction accuracy when utilizing multiple context sources with an alignment based prediction approach is shown,
- a method to determine which context sources will be useful to increase the prediction accuracy is given,
- the effect of stability against disturbances when multiple context sources are used is shown
- as also a survey on the practicability of a daily Smartphone use for context collection in terms of battery life is presented.

Zusammenfassung

Heutzutage sind wir umgeben von technischen System und Geräten die unser alltägliches Leben unterstützen und vereinfachen sollen. Wir nutzen Mobiltelefone, um überall auf Informationen zugreifen zu können, intelligente Haussteuerungen, um unsere Sicherheit und unseren Wohnkomfort zu steigern und Fahrerassistenzsysteme im Automobil, um unsere Fahrten angenehmer und sicherer zu gestalten. Allerdings erwarten wir oft ein vorausschauendes Verhalten solcher Systeme: das Mobiltelefon soll die Bahnstörung einblenden bevor wir die Fahrt beginnen, die Wohnung soll geheizt sein bevor wir sie erreichen, und das Fahrzeug soll den Verkehrsstau einblenden bevor wir in ihm stehen. Dafür beobachten diese Systeme unser Verhalten mit Sensoren, verarbeiten die Informationen mittels spezieller Algorithmen in Kontextquellen und leiten dann Kontextwerte daraus ab. Anschließend machen sie Vorhersagen zu künftig auftretenden Kontextwerten, oft nur auf einzelnen Kontextquellen basierend. Während wir dabei möglichst genaue Vorhersagen erwarten, ist der Vorhersagegenauigkeit ist jedoch eine Obergrenze gesetzt, die man basierend auf dem Informationsgehalt der Kontextwerte berechnen kann. Um diese Grenze nach oben zu verschieben, und damit genauere Vorhersagen zu ermöglichen, schlagen wir die Nutzung mehrerer Kontextquellen zur Kontextvorhersage vor. In dieser Arbeit beschreiben wir eine Möglichkeit verschiedene Kontextquellen zu kombinieren, zeigen den Gewinn der Vorhersagegenauigkeit, beschreiben eine Methode im Vorhinein festzustellen welche Kontextquellen sich besonders effektiv kombinieren lassen, zeigen den Gewinn an Störsicherheit durch den Einfluss von mehreren Kontextquellen und zeigen die Umsetzbarkeit in Bezug auf den Energieverbrauch eines ständig Kontextwerte erhebenden Systems.

Table of Contents

Publications

This is a list of publications I have done with my colleagues during the time I worked on the topic of this dissertation.

- König, I; Beau, P.; David K., "A new context: Screen to face distance", 8th International Symposium on Medical Information and Communication Technology 2014, Florence, Italy, 2. - 4. April, 2014
- König, I.; Klein, B. N.; David, K., "On the Stability of Context Prediction", AwareCast workshop, Ubicomp 2013, Zürich, Switzerland, 8 - 9 Sept., 2013
- König, I.; Memon, A. Q., David, K., "Energy consumption of the sensors of Smartphones", IEEE ISWCS 2013, Ilmenau, Germany, 27 – 30 August, 2013
- König, I.; Voigtmann, C.; Klein, B. N.; David, K., "Enhancing alignment based context prediction by using multiple context sources: experiment and analysis", 7th International and Interdisciplinary Conference Context 2011, Springer LNAI6967 Proceedings, Karlsruhe, September, 2011, pp. 159 – 172
- Lau, S. L.; König, I.; David, K.; Parandian, B.; Carius-Düssel, C.; Schultz, M., "Supporting Patient Monitoring using Activity Recognition with a Smartphone", Proceedings of the 7th IEEE International Symposium on Wireless Communication Systems (ISWCS), York, UK, 19th - 22nd September, 2010, pp. 810 – 814
- Lau, S. L.; Klein, B. N.; Pirali, A., König, I.; David, K., "Implementation of a User-Centric Context-Aware Playground", Workshop über Selbstorganisierende, adaptive, kontextsensitive verteilte Systeme (SAKS), KIVS 2009, Kassel, 5. March, 2009
- Lau, S. L.; Klein, B. N.; Pirali, A., König, I.; Droegehorn, O.; David, K., "Making Services Creation Possible for (Almost) Everyone", IST Mobile Summit, Stockholm, Sweden, 10 – 12 June, 2008

1. Introduction

Systems that adapt to a user's behaviour and support the user in his activities are called context sensitive. They recognize the context of the user by the help of sensors in the user's environment, which get influenced by the user's behaviour. In this work, we use the term *context* for a certain type of context like the *activity* or the *location*. Furthermore, we use the term *context value* for a certain value the context may have like *'sitting'*, or *'walking'* or *at 'home'*. The Term context value is also used in a similar way by Chen et al. in [10]. However, in the literature the actual value and the certain kind of context is often not separated [32] [15] [13] [12]. We need this separation to investigate the influence of the combination of multiple context sources.

After the data is captured by the sensors the context values have to be derived. This is usually done by feature extraction and classification [14]. We use the term *context source* to describe the feature extraction and classification steps after the sensor data has been gathered. The separation of the two steps is not necessary for the discussion in this work and we also want to combine several feature extractions and classifications, therefore we use the simpler term context sources. Context sources continuously deliver context values. The activity context for example can be derived with the accelerometer of a Smartphone. It can take context values such as 'sitting', 'standing' or 'walking' [4]. Such context values can then be used to adapt a system. An example for an adaptive system is the smart home. A smart home is a building where sensors sense the activities of its users and devices get controlled to support the activities of the users. If a user enters a room in the smart home the lighting can be switched on and adapted to the activity of the user to gain comfort. If the same user leaves the room the lighting can be switched off to save energy. However, in some situations a reactive adaption is not sufficient.

In some use cases an adaption of a system is needed before a certain context value occurs. This is needed if it takes some time for the system to achieve the desired state. In the smart home example from above the heating or cooling of a room must be started before a user enters this room. If the heating or cooling is only started when the user already enters the room, the room will be uncomfortably tempered. If the heating or cooling is started before the user enters the room the room will be comfortably heated when the user enters. Therefore, a prediction of upcoming events may be needed by adaptive systems.

Context prediction algorithms predict future upcoming events. Prediction algorithms use the repetitive behaviour of users'. The repetitive behaviour can be found in repetitive pattern in the context history. A context history is a time series of all past context values of a context source. The prediction algorithm learns repetitive pattern from the context history. In the next step the prediction algorithm compares the recently occurred context values, including the current context value, with the pattern in the context history. If a similar pattern is found, the context value following the similar pattern in the history can be used as a prediction. Afterwards the predicted value can be compared with the actually occurring value to determine the prediction accuracy.

A high prediction accuracy is important for the acceptance of proactive systems. A wrong prediction can cause an uncomfortable, unsecure or energy wasting system. Let us consider the smart home example once again. If the user's position context is predicted wrong, the room will be air conditioned without a usage or the room will be used without being air conditioned. This will be either a waste of energy or cause an uncomfortable situation. Both situations will reduce the acceptance of a proactive system.

One approach to increase the prediction accuracy is the use of better prediction algorithms. A number of different prediction algorithms like ARMA[13][14], Neural Networks for context prediction[15], Eigenbehaviors for context prediction [17], Alignment for context prediction [12] or HOSVD [16] can be utilized. However, despite all improvements there exists an upper boundary of prediction accuracy which is called predictability[23]. The predictability can be calculated for each user based on his individual context history. Prediction algorithms utilize repetitive pattern. The number of repetitive pattern in a context history strongly relies on a user's behavior. It might be difficult to predict context values for a spontaneous user who often does things he has never done before. On the other hand, it might be easy to predict context values for a user doing the same things every day and in the same order. The number of repetitive patterns limits the predictability of a user and is therefore used to calculate this upper limit of prediction accuracy. Although it is not possible to overcome the upper limit of prediction accuracy we propose another method to increase the prediction accuracy of a system.

We propose the to increase the prediction accuracy using additional context sources. Additional sensors can be used for additional context sources which then can be used by a prediction algorithm. We call this a multi context source prediction approach. Additional information related to a user can be gathered using additional sensors. The additional information may include previously not recognized pattern so the predictability is likely to be higher, when multiple context sources are used and therefore the prediction accuracy can be increased.

A multitude of sensors can be used more easily, since the number of sensors in our environment has increased over the last years. A Smartphone for example has a vast number of sensors like an accelerometer, gyroscope, compass, barometer, proximity sensor, GPS, light intensity sensors and within newer models also a pulse sensor. Smartphones are also equipped with multiple cameras and microphones. And they also have internal data sources like a user calendar, sent and received messages and emails or the list of phone calls. This internal data sources can also be used to derive context values. The Smartphone also has the advantage that it is worn close to the body of a user and therefore the collected sensor data strongly correlates with the behaviour of the user. Recently the Smartphones have been enhanced by wearables like smart watches or fitness trackers. Such devices are also equipped with sensors gathering information on the user’s behaviour which is often forwarded to a connected Smartphone. In addition to mobile sensors also mounted sensors can be found in our environment more often. Buildings get equipped with intelligent devices to enable them as smart spaces or smart homes. Often sensors are installed in windows or doors to gather their status, rooms are equipped with temperature, air quality and presence sensors, and even furniture can be equipped with sensors to notify the smart home about their usage. This variety of available sensors encourages their use for improving the context prediction accuracy.

The use of multiple context sources for context prediction raises several questions, which get answered in this thesis. These questions are: Can the prediction accuracy be increased using multiple sensors? How should a prediction algorithm work, to make use of several context sources? Which of all available context sources should be combined to increase the prediction accuracy? Can the redundancies of multiple context sources be used to make the prediction more stable against disturbances? Is a Smartphone useable for a daily context survey terms of battery life? Each of these questions is answered in a single chapter of the thesis. But first we are going to have a look on some motivating projects and the state of the art, where the upper boundary is also explained in more detail.

2. Motivation and Projects

This section reflects my personal motivation to research in the field of context and context prediction.

When I started working in the university I worked in an EU funded project named SPICE (Service Platform for Innovative Communication Environment) [49]. I worked in the service creation part of the project. Most of the innovative services we created relied on the context of a user. Usually the location context was used to trigger actions or to configure services. The influence of context was also a key element of the next project I worked in: MATRIX (unified context sensitive middleware for internet based tele medical services) [50], funded by the German BMBF (Federal Ministry of Education and Research). Context was used in this project to assist a tele medical patient monitoring and to support medical examinations. The prediction and use of future context values was also discussed in this project to find abnormalities in the behaviours of tele medical monitored patients. The actual project I work in named EnKonSens (Energy self-sufficient mobility for context sensitive building automation) [51] focuses on the reduction of energy usage in daily life, while at the same time the self-sustained life at home should be enabled as long as possible. Context is also the key for this project. It is used to adapt an automated building to the users' activities and to provide a safe environment, especially for elderly people. Context is also used in the project to adopt user interfaces to enable even non-technical users to control building automation systems.

During my work, I also had the chance to build up a laboratory environment with my colleagues, where we collected a wide variety of sensor data to derive all different kind of contexts. We equipped rooms with sensors such as temperature, humidity, light intensity and door and window sensors; we equipped cupboards with switches to see whether doors are open or closed; we connect all kinds of appliances to sensors to monitor their usage and we even equipped chairs with pressure switches. We also used a lot of mobile devices for the aggregation of sensor data, although the Smartphone is the most often used device, as it is equipped with many sensors. During the work with Smartphones I also proposed an approach to derive a novel context with the front camera of a Smartphone: the screen to face distance of a Smartphone user [52].

The work in the different projects as also the work in the laboratory environment at the university chair encouraged my motivation to research in the field of context prediction.

3. State of the art

The state of the art chapter consists of four parts. First we will have a look at the definition of context and some project and research work concerning context. Second we will get an overview on context prediction algorithms and frameworks. Third the stat of the art concerning the predictability is presented. We will need this in chapter 6 when we investigate a method of how to choose context sources that benefit from each other when combined. And finally, a state of the art concerning the energy consumption of Smartphone sensors is given. This is need for chapter 0 where we investigate the usability of a Smartphone for a daily sensing and context survey.

3.1 Definition of Context

A very famous description of a ubiquities environment in the field of context aware computing is the vision of Mark Weiser in his article "The Computer for the 21st Century" [21], published in 1991. He describes a world where computers are ubiquities, appliances automatically adapt to the users' needs and sensors are everywhere to keep track of the users' activities. In his vision the computers are getting smaller; the sensors are attached to the computers so they can derive the users' contexts and appliances are controlled by the computers. This implies the use of the context of the user by the computers. Later in 1999 Dey gave a useful definition of context which is:

"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 the application themselves." [32].

However, there are other definitions such as from Schilit et al. [33] who defined context categories, Lieberman et al. [34] who limits context to input and output data from a computer or Fitzpatrick et al. [35] who refined the definition of Dey by introducing the terms sensor and actuator in context. In this thesis, we will use the definition given by Dey.

Much research has been done in the field of applications and systems reasoning context and reacting on context, so called context aware systems. In this work context derived from permanently installed sensors and from mobile sensors is used. In accordance to this approaches are selected for this state of the art section where permanently installed sensors as also mobile sensors are used. First some research work concerning permanently installed context aware systems is described. Afterwards an overview on portable context aware systems is given.

One of the most prominent examples for context awareness used in buildings is the Neural Network House from the University of Colorado [42]. The house was equipped with sensors like temperature sensors, illumination sensors, sound level sensors, motion detectors, status sensors of windows and doors and the status and setting sensors off all appliances. The system could control the air heating, the lighting, the ventilation and the water heating. The focus of this house was to implement a learning appliance control system. Likewise, the MavHome [40][41] from the Washington State University and the University of Texas at Arlington was also equipped with similar sensors and appliances. However, the focus of the MavHome was to build an agent based control system for the house. Smart doorplates were installed in an office building at the University of Augsburg [45][46]. The doorplates could direct office visitors in the absence of a user and display the office presence context. To derive the office presence context identification tags were used. Researchers from the Lancaster University installed load sensors in different surfaces and floors [47]. Based on the load sensors the position context of objects as well as objet interactions could be derived. The smart home utilized in this work was equipped with presence detectors (PIR sensors), door sensors (open/closed) and the status of all light switches and several appliances were available. Although we decided to only use the presence detectors of each room. This was done to focus on the prediction accuracy and not on the improvement of the context reasoning.

Gellersen et al. investigated the use of multiple sensors to discover multiple situations with a small portable device[48]. The device was equipped with sensors like a microphone, light sensor, accelerometers, air pressure sensors and certain gas concentration sensors. It may be interesting to recognize that the TEA device had similar dimensions to a mobile phone, and most of the mounted sensors are today integrated in modern Smartphones. The described approach also focuses on the use of a multitude of sensors as it is also done in this work. They also propose to use a multitude of sensors to increase the prediction accuracy. However, they did not show the influence of several sensors on the prediction accuracy and the also did not give a method of how to select different sensors to increase the prediction accuracy, which is done in this work.

Van Laerhoven et al. developed a wrist worn device [43] to log motion data over a long-time period. Their goal was to derive several contexts with imprecise but energy efficient sensors. Although the idea of a wrist worn sensor was not novel to the time the device reminds of today’s smart watches and fitness trackers, where also sensors are worn at the wrist. Based on the idea of a long time worn device Lau et al. developed a Smartphone based system to derive the activity context of a user [4]. Therefore, the accelerometer of a Smartphone was used to derive the activities *sitting*, *walking* and *standing*. This system was also utilized in a telemedicine project to support patient monitoring and medical emanations [44]. Another usage of the Smartphone sensors was shown by Ye et al. when they used the Smartphone barometer to determine the floor level context of a user [39]. In this work the approach of Lau for the activity recognition is used. Although the barometer approach described by Ye has been implemented and user in this work.

3.2 Context prediction

In this subsection, several different context prediction approaches are described. The description is not a detailed technical explanation of each approach but rather a description of the idea of each and most important, the relation to this work is given.

Mayrhofer developed a context prediction framework. For this framework, he implemented and tested seven different context prediction algorithms, namely ARMA, Backpropagation Multi-Layer Perceptron, Support Vector Regression, Central tendency, Active LeZi, Hidden Markov Modell and Support Vector Machines [13][14]. The framework uses different sensors to derive features which serve as context values. Therefore, the feature extractors in his framework correspond to the context sources in this thesis. The advantage of his framework is a ready to use software form multiple platforms. It was designed to ease the use of context and context prediction for programmers. The framework builds vectors out of the values from different context sources which are then input data for the predictors. With this approach, he also uses a multitude of sensors for his prediction. However, he never investigated the influence of multiple context sources on the prediction accuracy. He also never investigated on an algorithm on how to choose which context sources should be combined to increase the prediction accuracy.

Sigg proposed the use of sequence alignment for context prediction [12]. The alignment algorithm was originally designed for use in bioinformatics, to find DNA matchings. Its strength is finding subsequences which match only partially, including gaps or mismatches. This way similar context value pattern can be found, even if the pattern generating user behaves slightly different over time. This approach is very beneficial when using non-ordinal context values (like 'sitting', 'walking', 'standing') and when a repeated behavior pattern is similar to a past time behavior pattern but not exactly equal. Sigg proposed a method of using multiple context sources in parallel for prediction. However, he did not investigate the influence of multiple context sources on context prediction. In chapter 5.1.2 we also investigate his proposed method of multiple context sources and enhance it to a prediction approach with higher prediction accuracy.

Petzold investigated prediction methods for the upcoming user locations, especially in an office building equipped with smart doorplates [56]. Therefore, he implemented a State Predictor [18] and a Bayesian Network [15] to predict future whereabouts of users. Vintan was investigating also on the smart doorplate experiment from Petzold, by implementing a Neural Network predictor [19]. All three approaches only focused on the use of location data as context source. Thus, they did not investigate the influence of multiple context sources on the location prediction.

Eagle proposed the use of Eigenbehaviors for context prediction [17]. With his approach, he focuses on the prediction of the user location. The location data was collected with multiple sensors and afterwards labeled with a Hidden Markov Model. This way he also used only a single context source for context prediction.

Voigtmann proposed a method he calls Collaboration-based Context Prediction [16]. This method is based on the Higher Order Singular Value Decomposition technique, which is also used in content recommendation systems [20]. His system can find similarities between users and therefore predict the context values of one user based on the history of a similar user. In his investigation, he uses only one context source but from different users. However, he did not investigate the influence of one context source for the prediction of another context source.

Some of the above described research is using multiple context sources while some research focuses only on a single context source. Most of the presented research work is utilizing multiple sensors to reason the context values more accurately. However, none of the research work investigates the influence of multiple context sources for context prediction. This encourages this thesis in which we investigate the influence of multiple context sources on the accuracy, the stability and the energy usage of context prediction, and where we also show how multiple context sources can be combined.

3.3 Predictability according to Song

The context prediction approaches presented in the last section vary in their prediction accuracy. This is caused by different factors such as the utilized prediction algorithm, the utilized dataset, and the parameter settings for the algorithms. With the optimal parameter settings, any of the described prediction algorithms may gain some prediction accuracy. However, the prediction accuracy has an upper limit, which we call the predictability. This upper limit is not relying on the utilized algorithm but on the utilized dataset. Each dataset has its own specific predictability.

The relation between a time series dataset and the predictability was already introduced in 1992 by Feder et al. [22]. In 2010 Song et al. [23] defined the predictability by the use of Fano's inequality in the field of location prediction. This predictability was later modified by McInerney et al. [24] in 2012 when they proposed to use only a section of the history. Song used the whole history for the calculation of the predictability, but they found periods of low predictability in daily life. Based on this Baumann et al. proposed in 2013 to use instantaneous entropy, to determine the predictability [25]. The instantaneous entropy only takes a limited time section and not the whole history into account, when calculating the entropy. Thus, he describes a momentary predictability. In 2014 Smith refined this limit by considering other constrains like the spatial resolution when using location data [26]. We decided to use the definition given by Song as we not only use location data and we also want to compare several combinations of context sources independently from the chosen time section of the data. Therefore, the next subsection will give a more detailed description on the predictability introduced by Song.

The predictability derived by Song et al. relies on Fano's inequality and the entropy rate of a context source. Fano's inequality is from the field of information and coding theory. It connects statistical features, especially the entropy rate of a symbol source with the error probability of the next symbol from the source. This is used to verify transmissions in the field of communication technology. Song et al. applied this to the field of context prediction, when he wanted to know how well a future context can be predicted, given the entropy rate of a context source is known.

Song considers a context source as a stochastic process generating random variables. This enables him to calculate the corresponding entropy rate. The entropy rate is a measure for the uncertainty of a context source. Intuitively, a source with a high uncertainty allows only a low predictability and vice versa. Therefore, he used the entropy rate of a context source with Fano's inequality and derive a formula for the predictability. The derived equation for the predictability is the following:

$$\mathcal{H}(\mathcal{X}) = -\Pi^{max} \log_2 \Pi^{max} - (1 - \Pi^{max}) \log_2 \frac{1 - \Pi^{max}}{N - 1} \tag{1}$$

$\mathcal{H}(\mathcal{X})$ is the entropy rate of a context source, N is the number of different context values from a context source and Π^{max} is the predictability.

The entropy rate $\mathcal{H}(\mathcal{X})$ has to be calculated for each context source. It is defined as

$$\mathcal{H}(X) = \lim_{\mathrm{n}\to\infty} \frac{1}{n} H(X_n | X_{n-1}, X_{n-2}, \dots, X_1) \tag{2}$$

On the right side of the equation a new entropy occurs: the conditional entropy $H(X_n | X_{n-1}, X_{n-2}, \dots, X_1)$. The conditional entropy is different to the normal entropy when it conditions the occurrence of a certain time series of context values $X_{n-1}, X_{n-2}, \dots, X_1$before the context value X_n. Considering the past context values when calculating the information of a context value is important. A lot of information is in the order of the context values in a context history. This is because our actions are, to some degree, relying on each other. If we assume the context to be the *user activity* context for example, we can have a look at the three context values '*sitting*', '*standing*' and '*walking*'. It is very unlikely for a human to start walking after he was sitting. Usually he has to stand up, which will always generate the context value '*standing*' after '*sitting*' in his context history. This also occurs in larger scales, like leaving the house in the morning, going to work, coming back at the evening or in annual scale like making holidays every summer or visiting the family each Christmas. To make use of this available information in the concatenation of the context values the conditional entropy has to be used.

The conditional entropy must be calculated recursively, which is very computation intensive. To decrease the needed computation power an entropy estimator can be used. An estimator typically considers only a subpart of all possible symbol combinations while the estimation still converges to the real entropy value. This way a lot of computation power and time can be saved. However, an entropy estimator can only be used if the process generating the symbols is ergodic. This means the probabilities of the symbols and their combinations will converge against certain values. In the area of context prediction this means a user has to have a steady behaviour over time. Song et al. chose to use the Lempel-Ziv estimator because it "is known to rapidly converge to the real entropy of a time series" [27]. This estimator is defined as

$$H^{est} = \left(\frac{1}{n}\sum_{i=1}^{n} \Lambda_i\right)^{-1} \ln n \tag{3}$$

n is the number of elements in the context history and Λ_i is the length of the shortest time series from position i on in history which does not appear before i.

To calculate a value for the predictability Π^{max} we can use (1). The left side can be estimated by the use of (3). As equation (1) cannot be solved for Π^{max}, a numerical solver has to be used to calculate the numerical for Π^{max}.

3.4 Energy consumption of Smartphone sensors

In this section literature is presented concerning the energy consumption of Smartphones or components of Smartphones.

In 2012 Carroll et al. made an analysis of the energy consumption in a Smartphone [28]. In this study, they investigated the influence of the GSM, CPU, RAM, Graphics and LCD Backlight. They also had a closer look at the battery lifetime, according to different usages of the Smartphone, like SMS, Video, Audio, Phone call, Web browsing and Email.

Different usages and the resulting Smartphone energy consumption were also investigated in the same year by Flipsen et al. in their study about the sizing of Smartphone batteries [29]. They made a comparison of the battery use, where the individual energy consumption of different Smartphones and the individual battery capacity were related to different Smartphone usages.

A study not concerning the usage but the components of a Smartphone was made by Perrucci et al. in their survey on energy consumption of single entities on the Smartphone [30]. They investigated the energy consumption of Bluetooth, Wi-Fi, 2G, 3G, CPU, Mobile TV, Display, Memory, Voice, Video and SMS. The study gives a very detailed description of the energy consumption of these entities. For Bluetooth for example they measured the energy consumption in the states: BT off, BT on, BT connected and idle, BT discovery, BT receiving and BT sending.

A component not investigated by G.P. Perrucci et al. is the GPS which was investigated by Pérez-Torres et al. They implemented a power aware Middleware [31] and concentrated on using a GPS in the Smartphone and determining the correlated battery runtime.

Nevertheless, none of the above-mentioned publications investigated the energy consumption of Smartphone sensors like the accelerometer, gyroscope, compass, barometer, light sensor, and proximity sensor. This is what we did to conclude whether a Smartphone useable for a daily context survey. The results are presented in chapter 0.

4. Multi context prediction[1]

In an environment where more and more technical devices work autonomously, we expect them to work well. Their actions should be a benefit for our workflow, comfort, security and health. And they should also help to decrease energy losses [9]. Examples are assistant applications like Google now or Siri on our Smartphones. They utilize the context of the user to support them at actual tasks [1]. To be useful such systems have to be as accurate as possible when deriving the user's context to take reasonable actions. Context prediction systems have to meet the same design rule: they have to predict as accurate as possible to gain a high user acceptance. The following example may illustrate this. A smart home equipped with a prediction system can set the right temperature, before a resident enters the room. This can be archived by the prediction of the room usage. The room can then be air-conditioned, according the resident's preferences. An inaccurate prediction of the room usage ends either in a comfort loss or an energy waste. An unpredicted room usage will cause a non-air-conditioned room, which means a loss of comfort. A predicted but never occurred room usage will cause an unnecessary air-conditioned room, which means an energy waste. Therefore, the accuracy of a context prediction system is crucial for the user acceptance. Predictions have to be as accurate as possible.

The accuracy of a context prediction system can sometimes be improved using a better prediction algorithm or the refinement of the existing prediction algorithm. Such improvements are adequate steps as often a few percent of prediction accuracy can be gained [13]. However, as discussed in chapter 3.3 there exists an upper boundary of prediction accuracy, which cannot be overcome, even by refining or the changing the prediction algorithm. To overcome this upper boundary, we propose another approach.

The accuracy of a context prediction system can be improved by the inclusion of additional information from additional sensors. The upper boundary of prediction accuracy is determined by the amount of information in a context history. By invoking

[1] Parts of this chapter have been published in [36].

additional sensors, more information can be gathered and therefore the prediction accuracy can be improved [36].

In this chapter the multi context prediction approach is explained in detail. First the structure of common prediction systems is explained and terms are defied which are used in this thesis. Afterwards the proposed extension is explained.

4.1 Context prediction system

The goal of a context prediction system is to predict future upcoming context values. We divide such a system in three components needed for this task: first the actual context values of a user, second a context history, and third specific prediction algorithm. This subsection explains each component in detail. Also, some definitions are given for these components.

4.1.1 Definition of Context

Some minor extensions of Dey's context definition have to be made, to keep the wording of this thesis stringent. In this work, we distinguish between a certain type of context and the value a certain type of context may have. This is explained in more detail in the next paragraphs. We extend the definition given by Dey the following way: "[A] Context [value] 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 the application themselves." [32]

4.1.2 Context source

Sensors are components which get influenced by a user's behaviour and capture a variety of physical quantities. Usually they convert the physical quantity they are designed to measure into an electrical quantity. The electrical quantity gets digitalized and can furthermore be represented by a numerical value. Although the sensors we refer to usually have strong correlations to the user's behaviour, it is important to recognize for later chapters that sensors are not only influenced by the behaviour of a user. Other physical processes may also influence them. The accelerometer of a

Smartphone for example is influenced by the user's movement. But it is also influenced by thermal noise, which is obviously not correlated to the user's movement. We call all such other influences not correlated to the user's behaviour disturbances.

To "characterize" the information provided by a sensor we use a component we call context source [10]. A context source generates context values for a certain context (type) based on the input of one or multiple sensors. Usually specialized algorithms fulfill this task. An example is the context called 'user activity' presented in [4]. The 'user activity' can generate context values like 'sitting', 'standing' or 'walking'. The presented algorithm derives these values from the accelerometer of a Smartphone worn in the user's pocket.

A context source generates a timely ordered series of context values. A new context value is generated depending on the settings of the context source, either when new sensor values arrive or after a certain timespan has passed. The context values can then be further used to trigger actions, predict future context values or be stored in a context history.

4.1.3 Context history

A context history is a database where past time context values are stored. This is often needed by context prediction algorithms [14]. Some Prediction algorithms need a history to learn context value patterns, other to lookup context value patterns.

All the above described terms are summarized in Table 1. Also, exemplarily values are given in the last column, where possible.

Table 1: Differentiation of terms used in this thesis

Name	*What it is*	*Examples*
Context	A certain type of context	'user activity' or
Contexts	A combination of multiple types of contexts	'user activity' and 'location'
context value	The value a certain kind of context may have	'sitting' or 'standing' or 'walking' or 'at home'
context values	A combination of multiple values a certain kind of context may have.	'sitting' and 'walking'

	Also a combination of multiple values and multiple contexts may	‘sitting’ and ‘at home’
context source	A source that generates context values from sensor values. Each context source generates a certain	Algorithm to derive the location or an algorithm to derive the user activity
context history	A database where past time context values are stored	
context prediction algorithm	The algorithm that predicts future upcoming context values	state predictor, ARMA [13][14], Neural Networks [15], Eigenbehaviors [17], alignment [12], HOSVD

4.2 Basic idea of context prediction algorithms

The basic purpose of a prediction algorithm is to predict upcoming context values. Therefore, the prediction algorithms utilize repetitions in human behaviour. When we perform a certain task we usually behave in a certain way. And when we repeat the same task another day we will repeat our behaviour. Of course, our behaviour will not be totally equal when we repeat a task but it will show similarities. Context sources which rely on our behaviour will produce a series of context values, which we call a pattern. A repeated task will produce an equal or at least similar pattern.

Context prediction algorithms learn or use pattern to predict future context values. First they use a history to learn the algorithm. Then they utilize a series or recently occurred context values, including the actual context value, and try to find this recently pattern in the history or in the data they have learned. Afterwards they predict the most likely following context value, based on what they have learned from the past.

Several prediction algorithms are available for the context prediction tasks. Examples are ARMA for context prediction [13][14], Neural Networks for context prediction [15], Eigenbehaviors for context prediction [17], alignment for context prediction [12], HOSVD for context prediction [16]. The all differ in the way they learn, their memory usage, their runtime, the type of context values they can process and their impact on privacy.

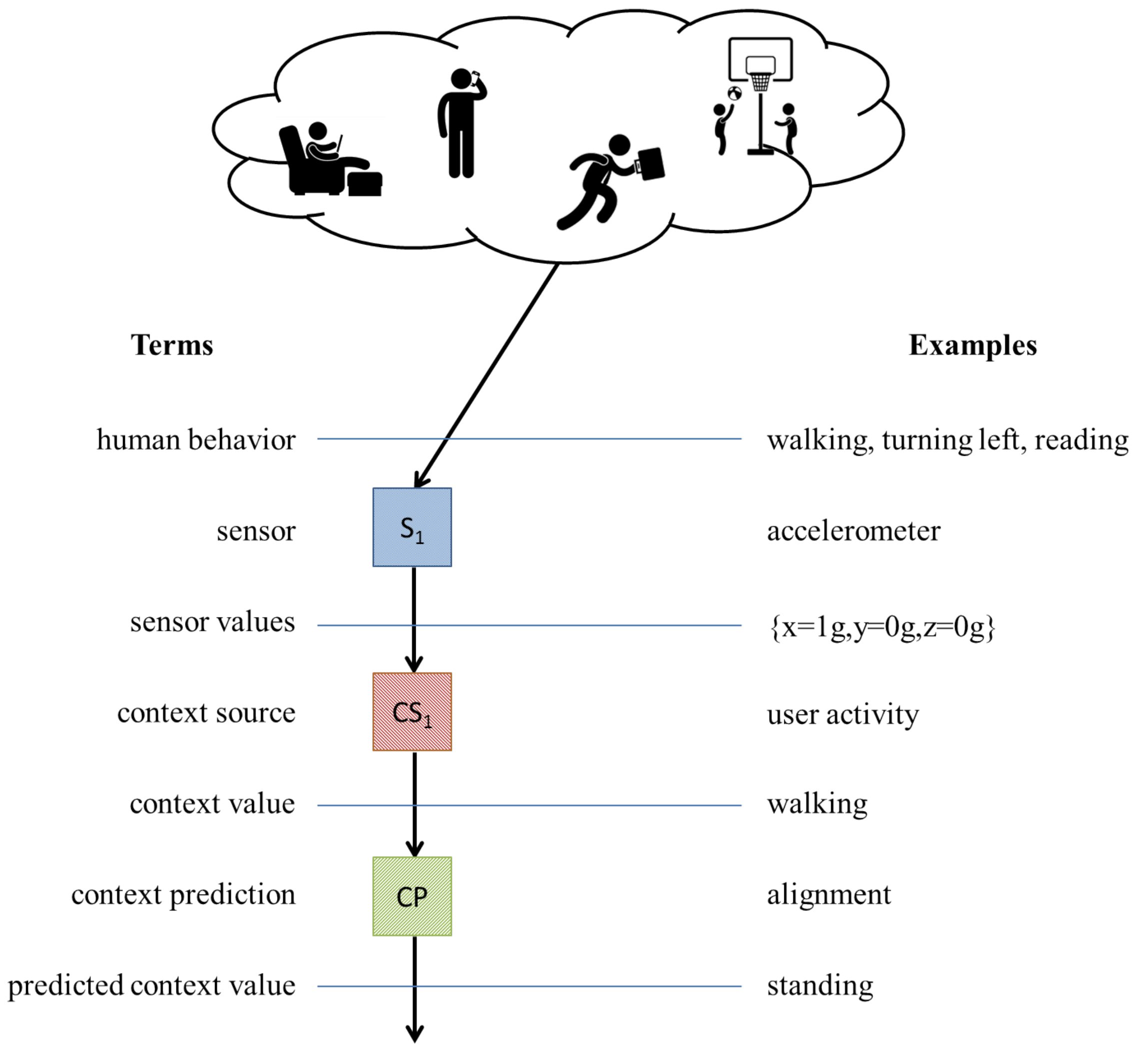


Figure 1: From human behaviour to context prediction

4.3 Basic idea of a multi context source prediction approach

Most context prediction algorithms are used to predict values for a certain context source, like the 'user activity' or the 'user location'. Our approach also aims this goal. But in difference to most prediction algorithms we utilize not only the information generated by this certain context source. We also utilize the information generated by other context sources [11]. The difference can be seen by comparing Figure 2 with Figure 1. Figure 1 shows the usual approach: a certain context source CS_1 is connected to a context prediction algorithm CP. Only information generated by this context source is utilized to predict upcoming context values for CS_1. In Figure 2 several context sources $CS_1 \ldots CS_n$ are connected to a context prediction algorithm CP. Still upcoming context values are only predicted for the prior chosen context source CS_1. But in contrast to the approach shown in Figure 1 also information generated by the other context sources $CS_1 \ldots CS_n$ are used to predict upcoming values for CS_1.

The proposed extension only works when two conditions are fulfilled. First: more than one sensor has to be available and second: the sensors have to be correlated to the user's behaviour to gain additional information. Both conditions are discussed in detail in the two following paragraphs.

Multiple sensors related to one user are often available. A today's Smartphone like the iPhone 6 for example hast a vast variety of sensors. It is equipped with an accelerometer, gyroscope, barometer, compass, proximity sensor, GPS, light intensity sensor, and one or more microphones. Also, the wireless communication possibilities of Smartphones can be utilized as sensors, like their signal strengths and address systems (GSM CELL ID, Wi-Fi SSID, and Bluetooth MAC address). Furthermore, the user related data stored in the Smartphone like the history of phone calls, send and receive messages and the user's calendar can be utilized as sensors. And if the Smartphone is connected to a wearable like the Apple Watch even more sensors are available, like for example a wrist worn accelerometer and a heart rate sensor [2]. Another example of a multitude of available sensors is the smart home. A smart home also offers a variety of sensors related to a user. Rooms can be equipped with presence detectors and furniture with usage sensors. An occupied chair or bed or an opened or closed cupboard provide information about the user's actions. Smart devices inside the home like smart refrigerators or smart ovens sense and communicate their usage. Smart meters can also serve as sensors for the usage of electricity, gas or water inside the smart home. The two exemplary domains Smartphone and smart home show that often multiple sensors are available, when the user action is observed by technical systems. Sometimes sensors from different domains can also be combined to gain more information [37].

The use of additional context sources and thus additional sensors only gains information when the different sensors are correlated to the user's behaviour. A context source connected to a sensor that is uncorrelated to a user's behaviour cannot add any information. For example: A Smartphone accelerometer can be used to derive the context 'user activity'. Of course, only the activity of the user who carries the measuring Smartphone can be derived. When we add a second context source connected to the accelerometer in a Smartphone of a second user, living apart from the first user, we will not gain any information from this addition. However, in most environments where we are able to sense a user's behaviour, we have several sensors, which are usually correlated. In the smart home, different sensors are influenced by the behaviour of one user. And in the Smartphone, several sensors are mounted in one case and thus usually several sensors get influenced at the same time by a user's behaviour. Accordingly, different sensors are influenced by one source, the user behaviour, and thus produce correlated information.

Not all context values added from one context source in addition to another context source are additional information. Some data may just be redundant. Some data may be useful to enhance the prediction accuracy and some may even be disturbing. This raises three important question: How exactly should a context prediction algorithm combine date from different context sources to improve the prediction accuracy? How can we decide which combination of context sources is useful and which combination just ads disturbances? And can we use the redundancies to keep the prediction stable against disturbances? These questions are answered in the next chapters.

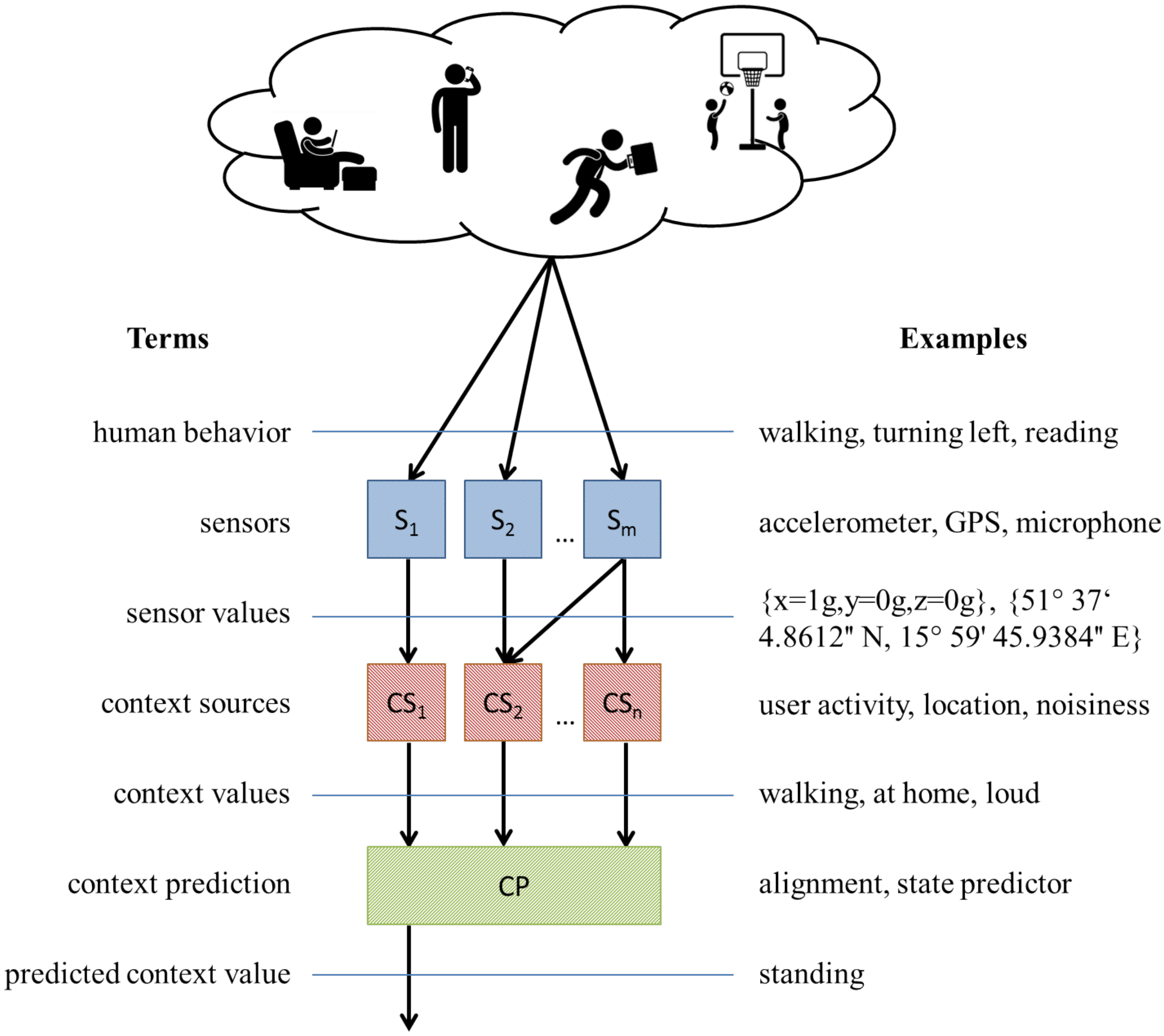


Figure 2: From human behaviour to context prediction with multiple context sources

5. Enhancing alignment based context prediction[2]

In the last chapter, we discussed the advantages of the use of multiple context sources for prediction. However, to verify the benefits in using multiple context sources by an experiment, we still need a concrete implementation of a prediction algorithm utilizing multiple context sources.

The author of [7] proposed a method of how to use alignment for the prediction with multiple context sources. His proposed method mainly uses a pre-mapping, explained in more detail later in this chapter. Therefore, we refer to this method as the mapping approach. Unfortunately, the mapping approach does not use all information available in the different context histories generated by the different context sources. Thus, we propose a different approach to overcome the disadvantages of the mapping approach.

In this chapter, we propose a novel method of how to use multiple context sources for context prediction. The approach utilizes the correlations between different context sources but does not expect the context sources to be entirely correlated. In parallel to the mapping approach our novel approach also utilizes alignment for context prediction. Therefore, we refer to this approach as the alignment multi context prediction.

This chapter first describes each approach. Afterwards the prediction accuracy of the approaches is compared analytically. In section 0 an experiment we performed to compare the two approaches as also the influence of multiple context sources on the prediction accuracy is described. Thereafter the results are given and discussed and finally a conclusion is drawn.

5.1 Alignment and multi context extension

This subchapter will explain how local alignment can be used for context prediction. Afterwards the two approaches extending the alignment to utilizing multiple context

---

[2] Parts of this chapter have been published in [36].

sources are explained. Graphics are used to illustrate the alignment process for each approach and also to highlight some problems regarding the mapping approach.

5.1.1 Alignment for context prediction

Alignment is originated in bioinformatics [8]. It can be used to measure the similarity of two sequences or to find a similar subsequence in a larger sequence. *Global alignment* is the alignment of two sequences of a similar length. It is used to measure the similarity of these sequences. *Local alignment* can be used to locate a shorter sequence inside a much longer sequence. These two applications occur when DNA sequences need to be compared to find affinities or when a DNA part needs to be located in a larger DNA sequence to find the location where the information is coded in the DNA.

Sigg proposed the use of alignment for context prediction [12]. Alignment is very suitable for the prediction of human behaviour because it is able not to only to find exact matching pattern but also to find similar pattern. Alignment recognizes a pattern still as similar, even if gaps have to be insert or if mismatches occur in the alignment. In human behaviour, we usually have similarities in our repetitive behaviour pattern, but we also have small fluctuations. Gaps and mismatches are illustrated by an example for each in the following two paragraphs.

Gap example: A user goes every morning from his *home* to the *bus stop*, to the *bakery* and then to his *work*. One morning, due to a lack of time, he just goes from his *home* to the *bus stop* and then to his *work*. He skips the *bakery* this day. Alignment is still able to align the sequence {'*home*', '*bus stop*', '*bakery*', '*work*'} with {'*home*', '*bus stop*', '*work*'} by aligning {'*home*', '*bus stop*', GAP, '*work*'} to it.

Mismatch example: A user goes every morning from his *home* to the *bus stop*, to the *bakery* and then to his *work*. One morning, due to a changed appetite, he goes from his *home* to the *bus stop*, to the *butcher* and then to his *work*. He exchanges the *bakery* with the *butcher* this day. Alignment is still able to align the sequence {'*home*', '*bus stop*', '*bakery*', '*work*'} with {'*home*', '*bus stop*', '*butcher*', '*work*'} by aligning {'*home*', '*bus stop*', MISSMATCH, '*work*'} to it.

For the prediction of future context values local alignment is used. Therefore, we need a history filled with many time ordered past context values. We also need some recently observed context values. This is illustrated in Figure 3. Each letter in the blue boxes represents a different context value. On the right side the recently observed context values can be seen. The context value of the empty blue box is the future value and has to be predicted.

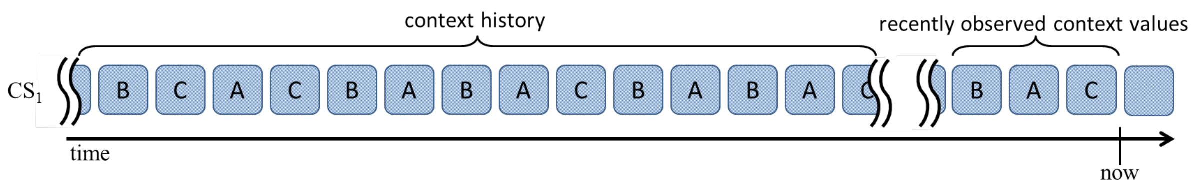


Figure 3: A context histories from context source CS_1, recently observed context values on the right

The local sequence alignment aligns the sequence of recently observed context values with the history. The goal of the alignment is to find a sequence that is as similar as possible to the recently observed sequence. Figure 4 illustrates an alignment at a certain position in the history. The light blue box behind the context values {'*B*','*A*','*C*'} is the position where the recently observed sequence {'*B*','*A*','*C*'} is aligned with the highest similarity.

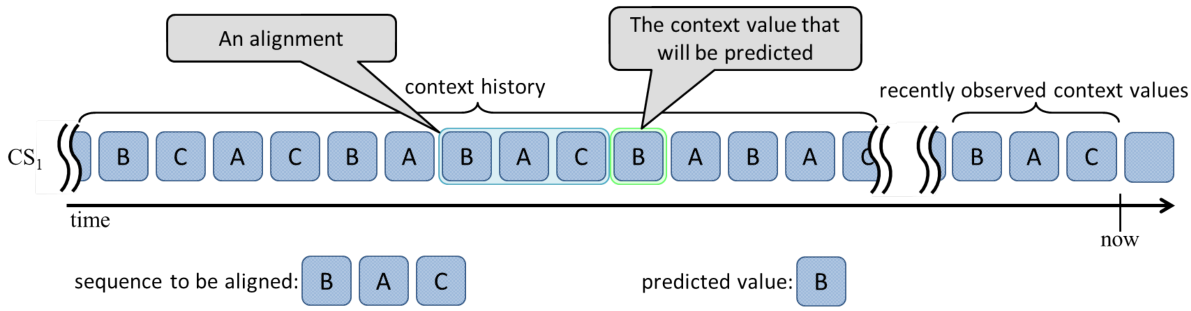


Figure 4: An alignment of some recently observed context values (light blue box) and a context value selected as prediction (light green box)

The context value following the position of the best matching alignment is taken as a candidate to predict the future (light green box in Figure 4). This is done under the assumption that users will behave the same way the already did. However, sometimes multiple alignments are found in one history. Then the prediction algorithm has to decide which context value has to be predicted. This can be done by a random choice between candidates or by a majority count of candidates. Also, other considerations may influence the decision like the age of a candidate. An old candidate may be less useful as the user's behavior may have changed recently.

Sequence alignment is not only able to align perfectly matching sequences. As described it can also align sequences with mismatches or gaps. This is shown in Figure 5 where the sequence in the history has an additional '*C*' in between. Therefore, a gap has to be insert between the '*B*' and the '*A*, *C'* in the sequence to be aligned. While this alignment has a higher alignment cost than the one shown in Figure 4, is shows also similar behavior pattern can be used for prediction.

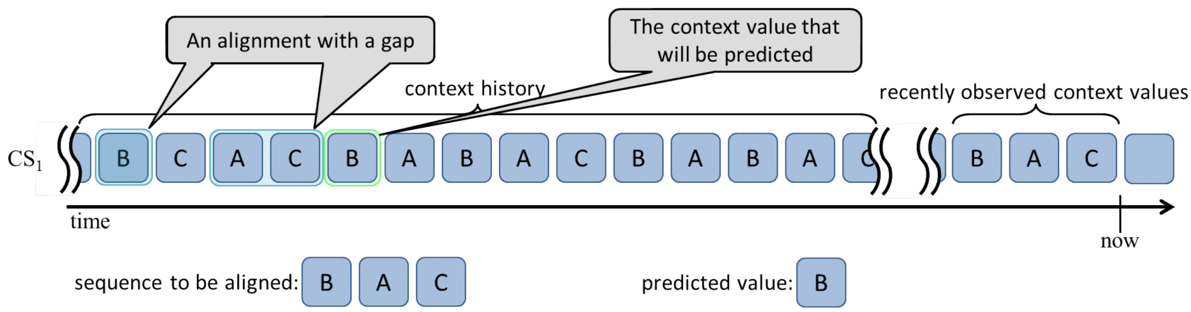


Figure 5: An alignment with a gap (light blue boxes) and a context value selected as prediction (light green box)

5.1.2 Enhancement to multiple sources with a mapping approach

In chapter 4 we discussed the advantages of a context prediction system, which is based on multiple context sources. In this subsection, we are going to have a look on two different approaches to achieve this.

The enhancement from one to multiple context sources also ads more histories, one for each added context source. Figure 3 shows the history for an approach where we use only one single context history. Figure 6 illustrates the new situation, where we have three exemplary context sources CS_1-CS_3. Each context source generates context values for his context history so that we now have three context histories. We choose the Latin alphabet for CS_1, hand signs for CS_2 and the Greek alphabet for CS_3, so we better remember that context sources usually produce nominal values and that we cannot compare values from two different context sources. On the right side of Figure 6 we see the recently observed context values from each context source. Nevertheless, the task keeps still the same: we want to predict the next context value for CS_1, indicated by the empty blue box in the upper right.

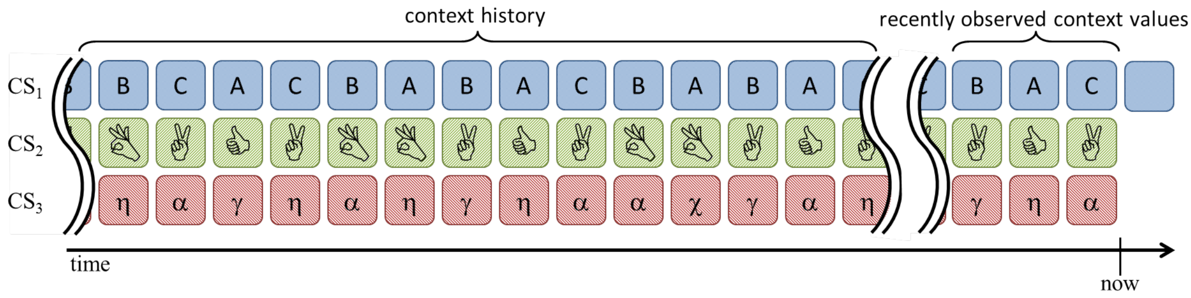


Figure 6: Three context histories from three different context sources CS_1-CS_3, recently observed context values are on the right side

One simple approach proposed by the author of [7] is to map all existing combinations of context values from the different context sources at a certain time into new context values. The mapping needs to be bijective; a certain combination of context values needs to be mapped into a certain new context value. This creates a new artificial context source, wherein all the previous context sources are combined.

Step one, illustrated in Figure 7: Mapping of all different context histories into one context history. Each of the yellow boxes is a new context value generated by a mapping of the original context values from CS_1-CS_3. All yellow boxes combined can be regarded as a context history generated by the artificially created context source CS_*.

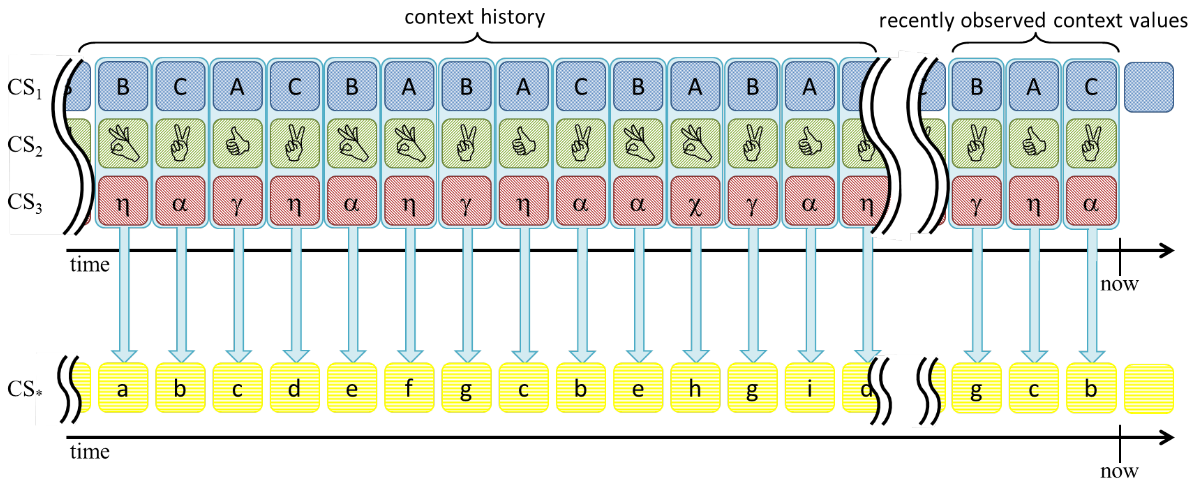


Figure 7: Mapping of context values into new context source CS_*

Step two, illustrated in Figure 4: A prediction can be made based on an already implemented algorithm. This also applies to the alignment for context prediction approach. The yellow row in Figure 7 is similar to the blue row of Figure 4, when we treat CS_* from Figure 7 as equivalent to CS_1 from Figure 4.

This approach has a major advantage: existing context prediction approaches designed for a single context source can be further used, even with multiple context sources. Only a mapping needs to be pre-processed. This is a big advantage as the context prediction algorithms do not need to be rewritten. However, this approach also has a major disadvantage in terms of information usage.

The mapping approach also has a major disadvantage: any disturbed context value from the original context sources will cause the mapping process to collect all disturbances and to ignore the information provided by the non-disturbed context sources. Note that we defined in chapter 4 any influence on the context sources which is not related to the user's behaviour a disturbance. This means a single disturbed context value and several non-disturbed context values generate a disturbed mapped context value. The information from the non-disturbed values can no longer be accessed. The mapping acts as a kind of disturbance collector.

The collection of disturbance is illustrated in Figure 8. The boxes with red marks represent disturbed context values. The first occurrence of a disturbance happens at the fourth column from the left. The mapping of {'C', disturbed, 'η'} generates a disturbed context value in CS_*. The information of the correlated occurrence of 'C' and 'η' is lost.

The information loss can also be clearly tracked in Figure 8 when we calculate the quote of disturbed context values before and after the mapping. Before the mapping we have 51 values in total and 5 are disturbed. This results in a disturbance rate of about 10%. After the mapping we have 17 context values and 5 are disturbed. This means a disturbance rate of about 30%.

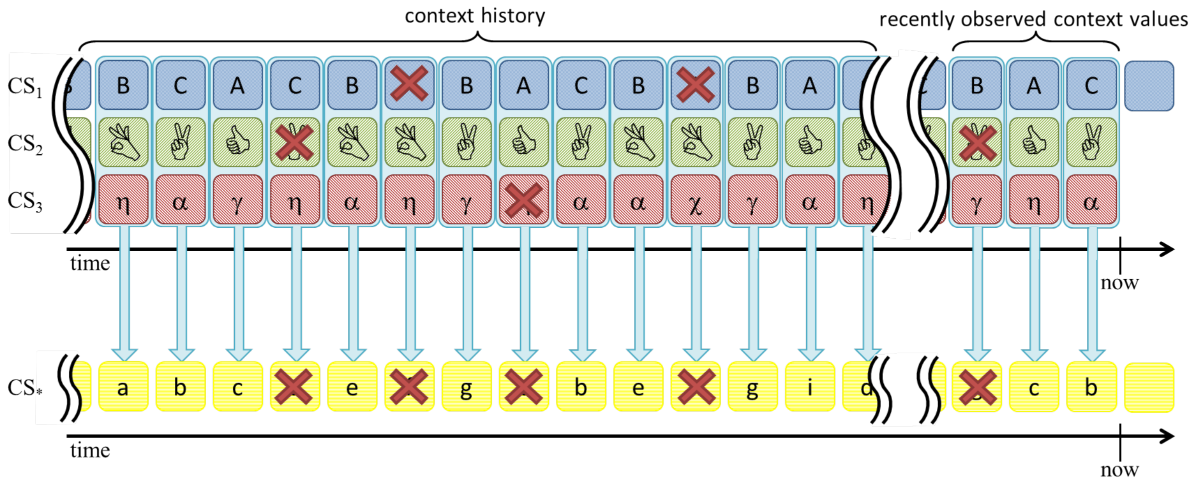


Figure 8: Wrong context values (red marks) are getting propagated through the mapping into CS_*

5.1.3 Multi alignment prediction

To overcome the disadvantage shown in the last subsection we propose another way of how alignment for context prediction can be enhanced to process multiple context sources. This is explained in more detail in this subsection.

The basic idea is to run the alignment several times, once for each context source. But in difference to the regular use of alignment for context prediction we take the value to predict always from the context source that should be predicted. This is illustrated with the help of an example. For the example, we consider again we use the three context source CS_1-CS_3 and we want to predict the next context value for CS_1. Each step is illustrated from Figure 9 till Figure 11.

Step one, illustrated in Figure 9: A prediction for CS_1 is made based on the history of CS_1. The recently observed context values {‘*B*’, ‘*A*’, ‘*C*’} of CS_1 are aligned with the history of CS_1, coloured in blue. The alignment is highlighted by the light blue box and the predicted value is highlighted by the light green box. This is still the same like the regular use of alignment for context prediction.

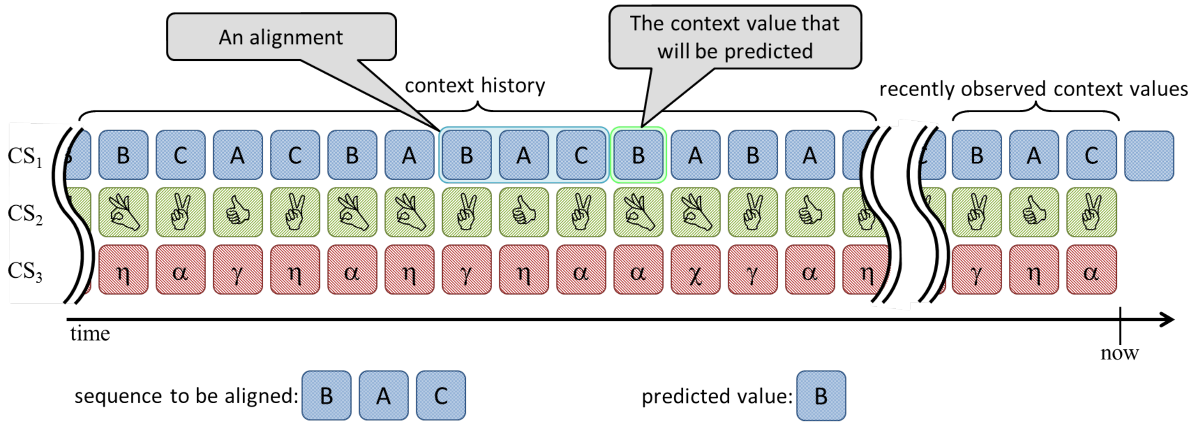


Figure 9: Alignment multi context prediction approach step one: a sequence of recently observed context values from CS_1 gets aligned with CS_1 and a prediction value is taken from CS_1

Step two, illustrated in Figure 10: A prediction for CS_1 is made based on the history of CS_2. The recently observed context values {'✌', '👍', '✌'} of CS_2 are aligned with the history of CS_2, coloured in green. (In this situation we have two alignments, highlighted by the two light blue boxes. This is only to show a situation where multiple alignments are made.) But in difference to the usual use of alignment for context prediction, we do not take the value to predict from CS_2 but we take it from CS_1. The position where we take the predicted value from is determined by the alignment in the history in CS_2, but the value is given by CS_1. We do this as we assume a correlation in the different histories between CS_1-CS_3. The two values taken to be predicted are highlighted by the light green boxes.

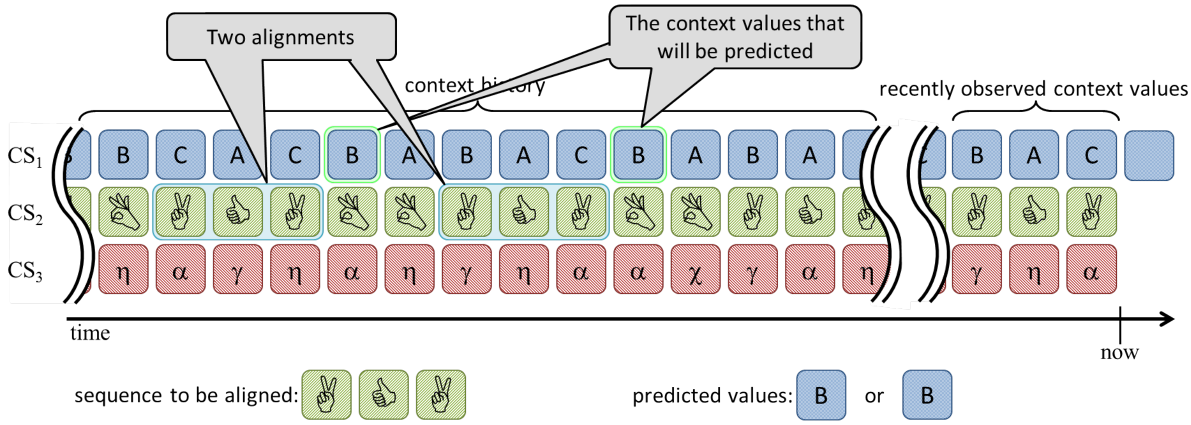


Figure 10: Alignment multi context prediction approach step two: a sequence of recently observed context values from CS_2 gets aligned with CS_2 and two prediction values are taken from CS_1. Note: although the sequence was aligned with CS_2, the prediction was taken from CS_1.

Step three, illustrated in Figure 11: A prediction for CS_1 is made based on the history of CS_3. The recently observed context values {'γ', 'η', 'α'} of CS_3 are aligned with the history of CS_3, coloured in green. (In this situation we also have two alignments, highlighted by the two light blue boxes. This is only to show a situation where multiple alignments with different results are made.) Here we also take it from CS_1. The position where we take the predicted value from is determined by the alignment in the history in CS_3, but the value is given by CS_1. The two values taken to be predicted are highlighted by the light green boxes. This time different context values {'*A*', '*B*'} are the result of the alignment.

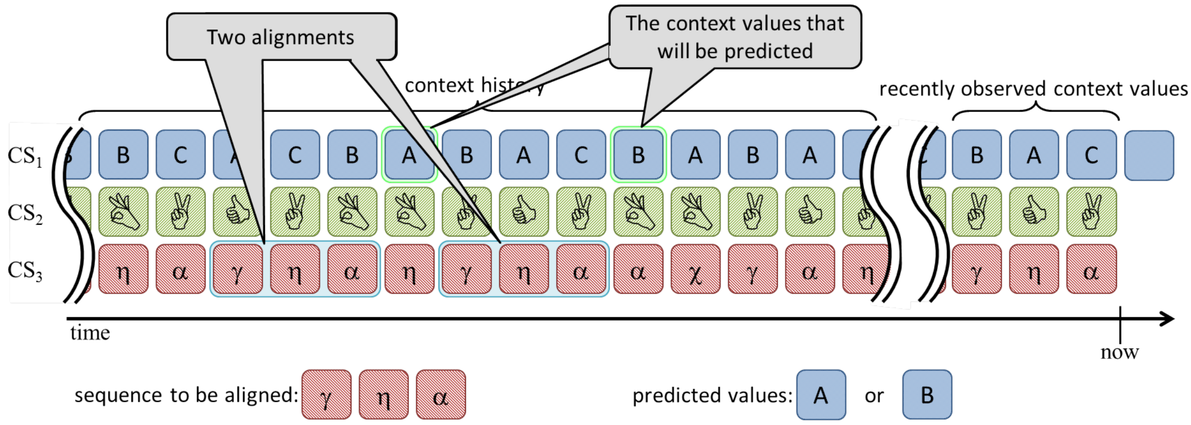


Figure 11: Alignment multi context prediction approach step three: a sequence of recently observed context values from CS_3 gets aligned with CS_3 and two prediction values are taken from CS_1. Note: although the sequence was aligned with CS_3, the prediction was taken from CS_1.

Step four: We have to choose between all results which context value we want to predict. The alignment in CS_1 resulted in '*A*', the alignment in CS_2 resulted in '*A*' and '*B*' and the alignment in CS_3 resulted in '*A*' and '*B*'. In our approach, we decided to always pick the majority. In the example, we would now predict '*A*' as future context value for CS_1. However, this can also be improved in future work by considering the correlation strength or other factors.

5.2 Analytic comparison of two multi context approaches

We decided to compare the two approaches in two ways: first by an analytic comparison of the prediction accuracy and second by an experiment. In this subsection, the analytic comparison is described.

The Idea of the analytic comparison is the following: we are going to derive for both, the mapping approach and the multi context approach, the probability of making accurate predictions. We show in which way the dependency of the accurate prediction relies on the probability of each single context value is without a fault. By doing this we can compare the two approaches afterwards.

We can calculate the probability the mapping approach predicts the accurate value is influenced by two factors: the prediction algorithm and the utilized history. We can assign both influencing factors a probability of being accurate. P_{pred} is the probability the prediction algorithm makes accurate predictions and $\bar{P}_{CS_*}$ is the average probability of the history of CS* being accurate. By multiplying these two factors we get the probability $P_{mapping}$ that mapping approach is accurate.

$$P_{mapping} = P_{pred} * \bar{P}_{CS_*} \tag{4}$$

The use of an average value is of course only a statistical way to determine the accuracy of a whole history. Some parts of the history may have a higher accuracy; some may have a lower. But after many predictions the average history accuracy will determine the average prediction accuracy. We calculate the average value $\bar{P}_{CS_*}$ by summing up all probability of the context values being accurate $P_{CS_*}(i)$, and divide by the number of context values in the history. The index i is an index to run over all context values in the history.

$$\bar{P}_{CS_*} = \frac{1}{n}\sum_{i=1}^{n} P_{CS_*}(i) \tag{5}$$

As described in the last subsection, CS* is an artificial context source, generated by the mapping process. Therefore, the probability of each element of the history of CS* being accurate $P_{CS_*}(i)$ relies on the context values it is mapped from. To calculate $P_{CS_*}(i)$ we have to multiply the probability of being accurate of each context value it is mapped from. The probability of being accurate of each context value from the original context histories is represented by $P_{CS_j}(i)$, where i is an index to multiply over all context values in the history , j is the index for the original context source and m is the number of context sources that are mapped into CS*.

$$P_{CS_*}(i) = \prod_{j=1}^{m} P_{CS_j}(i) \tag{6}$$

With this formula, we can already see what is happening during the mapping approach. $P_{CS_*}(i)$ will always be smaller (or equal) than the smallest $P_{CS_1}(i)$ ... $P_{CS_m}(i)$ it consists of. This is the in 5.1.2 described and in Figure 8 illustrated disturbance collection process of the mapping. Using (6) we can write

$$P_{CS_*}(i) \le \min\left(P_{CS_1}(i), P_{CS_2}(i), P_{CS_3}(i), \dots, P_{CS_m}(i)\right) \tag{7}$$

When we combine (7) with (5) and (6) we can also write

$$\bar{P}_{CS_*} \le \min\left(\bar{P}_{CS_1}, \bar{P}_{CS_2}, \bar{P}_{CS_3}, \dots, \bar{P}_{CS_m}\right) \tag{8}$$

where $\bar{P}_{CS_1}$ ... $\bar{P}_{CS_m}$ are the average probabilities of the according histories of CS_1 … CS_m are being accurate, which can be calculated similar to (5).

(8) is an important result: The history of CS_* generated with the mapping process is always less (or equal) correct than the original context histories of CS_1 … CS_m. With (4) we can conclude the average prediction accuracy follows this rule. This means the average prediction accuracy based on the mapping approach is always less (or equal) accurate than a prediction based on the original history. Thus, we cannot increase the prediction accuracy by adding multiple context sources when the mapping approach is used.

Now we can also have a look on the multi context approach to also relate its average prediction accuracy with the probability of each single context value being without a fault.

The multi context approach uses only context values which ae in the majority of all predicted values. Accordingly, we can calculate the average prediction accuracy P_{multi} by

$$P_{multi} = \frac{1}{elements\ in\ majority} \sum_{j=1}^{m} P_{pred1,j} * \bar{P}_{CS_{\mathrm{j}}} \Big|_{P_{CS_{\mathrm{j}}}(i+1)=majority} \tag{9}$$

This means the average accuracy of each used context histories $\bar{P}_{CS_{\mathrm{j}}}$ is summed up. It is only used when the prediction mad on it is part of the majority ($P_{CS_{\mathrm{j}}}(i+1) = majority$). To calculate the average accuracy, we divide by the number of elements in the majority. We always assume in the formula the prediction accuracy on the alignment $P_{pred1,j}$ for context 1, based on context j . However, we could also predict for another context without losing the generality.

We know an average value is always between (ore equal) the minimum and maximum values it is calculated from. Therefore, we can write with (9)

$$\begin{aligned} \min\left(P_{pred1,1} * \bar{P}_{CS_1}\Big|_{P_{CS_1}(i+1)=maj.}, \dots, P_{pred1,m} * \bar{P}_{CS_{\mathrm{m}}}\Big|_{P_{CS_{\mathrm{m}}}(i+1)=maj.}\right) \\ \leq P_{multi} \\ \leq \max\left(P_{pred1,1} * \bar{P}_{CS_1}\Big|_{P_{CS_1}(i+1)=maj.}, \dots, P_{pred1,m} \right. \\ \left. * \bar{P}_{CS_{\mathrm{m}}}\Big|_{P_{CS_{\mathrm{m}}}(i+1)=maj.}\right) \end{aligned} \tag{10}$$

The expression shows only some predictions are used in the calculation. To compare both methods we need to get rid of the conditions. To achieve this, we have look on two cases.

Case one: A not used single prediction has a lower probability of being accurate than all used predictions

$$\begin{aligned} P_{pred1,\zeta} * \bar{P}_{CS_\zeta}\Big|_{P_{CS_\zeta}(i+1)\neq maj.} \\ \leq \min\left(P_{pred1,1} * \bar{P}_{CS_1}\Big|_{P_{CS_1}(i+1)=maj.}, \dots, P_{pred1,m} \right. \\ \left. * \bar{P}_{CS_{\mathrm{m}}}\Big|_{P_{CS_{\mathrm{m}}}(i+1)=maj.}\right) \leq P_{multi} \end{aligned} \tag{11}$$

Now we can add the part left from the first inequality into the min function, as it will still be smaller than the original min function. This results in the loss of the conditions

$$\min\left(P_{pred1,1} * \bar{P}_{CS_1}, \dots, P_{pred1,m} * \bar{P}_{CS_{\mathrm{m}}}\right) \leq P_{multi} \tag{12}$$

Case two: A not used single prediction has a higher probability of being correct than the one with the minimal probability of the used predictions

$$P_{multi} \leq \max\left(P_{pred1,1} * \bar{P}_{CS_1}\Big|_{P_{CS_1}(i+1)=maj.}, \dots, P_{pred1,m} * \bar{P}_{CS_{\mathrm{m}}}\Big|_{P_{CS_{\mathrm{m}}}(i+1)=maj.}\right) \leq P_{pred1,\zeta} * \bar{P}_{CS_\zeta}\Big|_{P_{CS_\zeta}(i+1)\neq maj.} \tag{13}$$

Now we can add the part right from the second inequality into the max function, as it will still be larger than the original max function. This also results in the loss of the conditions

$$P_{multi} \leq \max\left(P_{pred1,1} * \bar{P}_{CS_1}, \dots, P_{pred1,m} * \bar{P}_{CS_{\mathrm{m}}}\right) \tag{14}$$

Now we can combine (12) and (14) and (4) and (8) to compare the multi context source approach with the mapping approach

$$P_{mapping} \leq \min\left(P_{pred1,1} * \bar{P}_{CS_1}, \dots, P_{pred1,m} * \bar{P}_{CS_{\mathrm{m}}}\right) \leq P_{multi} \leq \max\left(P_{pred1,1} * \bar{P}_{CS_1}, \dots, P_{pred1,m} * \bar{P}_{CS_{\mathrm{m}}}\right) \tag{15}$$

Now we can easily compare the probabilities of each approach of being accurate. The mapping approach accuracy will always be worse or equal but never be better than the multi context source approach.

5.3 Experimental comparison of two multi context approaches

After the analytic comparison, it would be nice to verify our findings in an experiment. To do this we designed an experiment to compare the two approaches, the mapping approach and the multi alignment approach. Therefore, we collected different datasets from both, data from a simulation and a real-world experiment. Each dataset consisted of the record of multiple context sources in parallel. Afterwards we split each dataset into a history and a test set. With the test set we performed a first series of prediction with both approaches and with one context sources. Afterwards we performed a second series of predictions with both approach but this time with two context sources. We calculated the prediction accuracies and compared them to each other to investigate the influence of multiple context sources on the two approaches.

The next subsections describe how the data was obtained and how the prediction accuracy was calculated.

5.3.1 Simulated Data

We used SIAFU, an open source context simulator [3] to generate the simulation data. We simulated an office environment, a scenario that is included in the simulator. In the simulation, different persons, called agents, moved through different offices. While the agents moved, different contexts were generated, depending on the time, movement and the position of the agents. In Figure 12 a screenshot of the simulator can be seen.

We chose to record the following four context sources: *activity*, *noise level*, *office area* and *Wi-Fi reception*. The *activity* context had one of the eight values '*AtDesk'*, '*EnteringToilet'*, '*Going2Desk'*, '*Going2Meeting'*, '*Going2Toilet'*, '*InTheToilet'*, '*LeavingWork'* and '*Resting*'. The *noise level* context had five possible context values '*average'*, '*loud'*, '*quiet'*, '*veryLoud'* and '*veryQuiet'*. The *office area* context indicated whether the agent was in an office or not. It was just represented by the context values '*true*' and '*false*'. The *Wi-Fi reception* context indicated the signal strength of the Wi-Fi at the agent's position, which had the five possible context values '*OutOfRange*', '*VeryWeak*', '*Weak*', '*Strong*' and '*VeryStrong*'.

Only the context sources *activity* and *noise level* were used in this experiment. Nevertheless, we decided to record two additional sources, as we also wanted to use the data in another experiment, described in 7.1.

Fourteen different agents were recorded and over 94000 datasets for each agent. We choose one agent out of the fourteen recorded as the results of the other agents showed the same trends and to keep the results section focused.

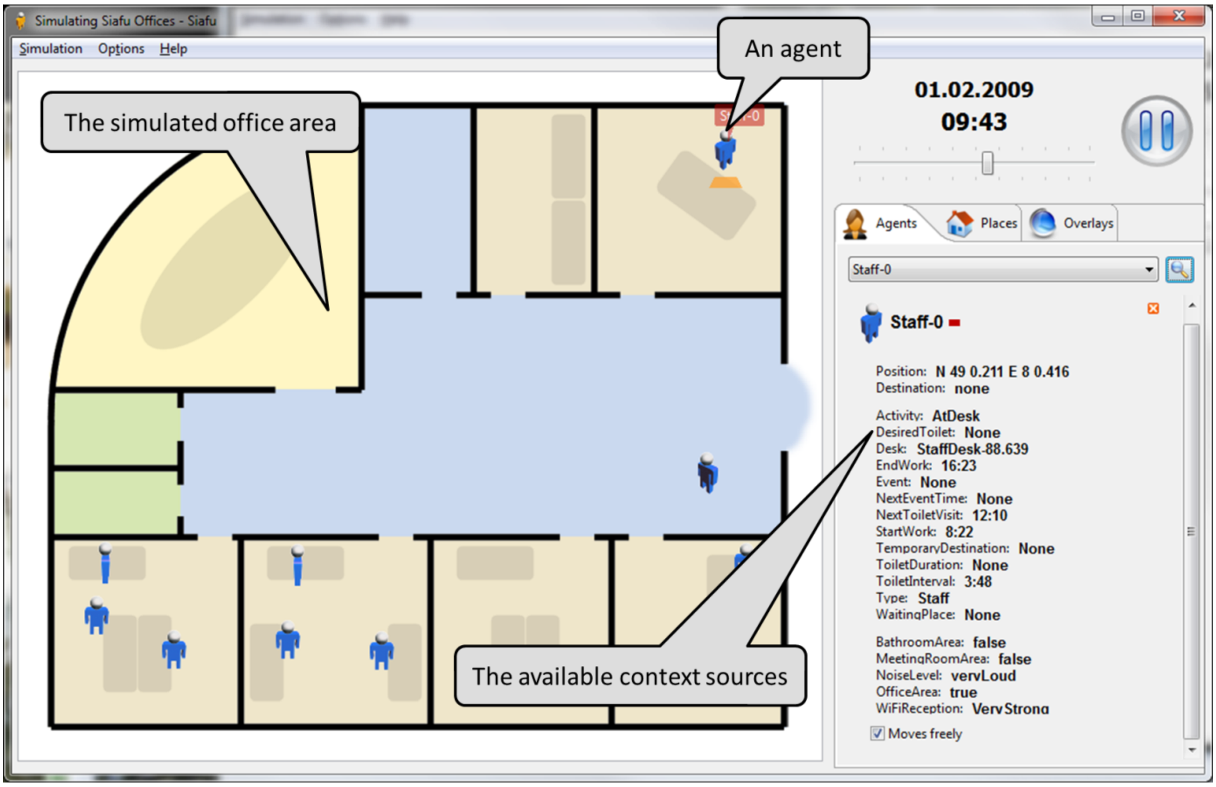


Figure 12: SIAFU, the context simulator used to collect the simulation data

5.3.2 Experimental Data

The location as well as the surrounding temperature of a user was logged, to collect the real-world data. We chose to log the two sensors location and temperature as they are usually correlated. Therefore, a user carried two devices in daily life over a period of eight days. Two devices had to be used as no Smartphone was available that could measure the surrounding temperature.

The location was collected by a Motorola milestone, an android based Smartphone. We activated google latitude on the phone, a former location recording service from google. Google latitude used GPS data, the Wi-Fi SSID and the GSM cell-id to determine the location of the Smartphone. After the collection was finished we downloaded the data, clustered it into 41 locations and labelled it like like '*home*', '*work*', etc. This dataset of timestamp annotated locations was used as *location* context source.

The temperature was collected by a SunSpot. A SunSpot is a little portable device from the company Sun which is equipped with a processor, some IO-ports and sensors, also including a temperature sensor, memory and an accumulator. The sunspot was carried together with the Smartphone by the user, while the temperature sensor was pointing away from the user to minimize the influence of the user's body temperature. We recorded the temperature each second and the data was downloaded during the night while the device was recharged. After the data collection was finished we post processed the temperature values, so our context source *temperature* had the three possible context values '*outside*', '*inside*' and '*unknown*'

5.3.3 Measurement of prediction accuracy

After the data was collected we made a series of test to determine the prediction accuracy, first with the mapping approach and second with the multi context source approach. We used a two-step procedure with both, the simulation data and the real-world data.

Step one was to make a series of predictions. Therefore, the datasets had to be split in to the history and the test set. The test set was about 1/3 of the collected data and the history was about 2/3. A small time series was cut from the test set and both approaches had to predict on this time series on the context history. Afterwards the next small time series was cut from the test set and another prediction was performed. This procedure was repeated until all data of the test set was used.

In step two the prediction accuracy was calculated. Therefore, each prediction was compared to the context value following the small time series from the test set which was used for the prediction. If the two symbols were equal the prediction was accurate, if not the prediction was wrong. This was done for all predictions and the average was calculated.

5.4 Results of the experiment

In this section the results of the above described experiment are presented. First the results of the prediction accuracy measurements based on the simulation data and afterwards the result based on the real-world data are given.

5.4.1 Simulation data results

The results from the experiment based on the context simulator are shown in Figure 13. When only one context source is used, both approaches perform equally. When a second context source is added, the mapping approach loses 18% of prediction accuracy, while on the other hand the multi alignment approach gains 4% of prediction accuracy.

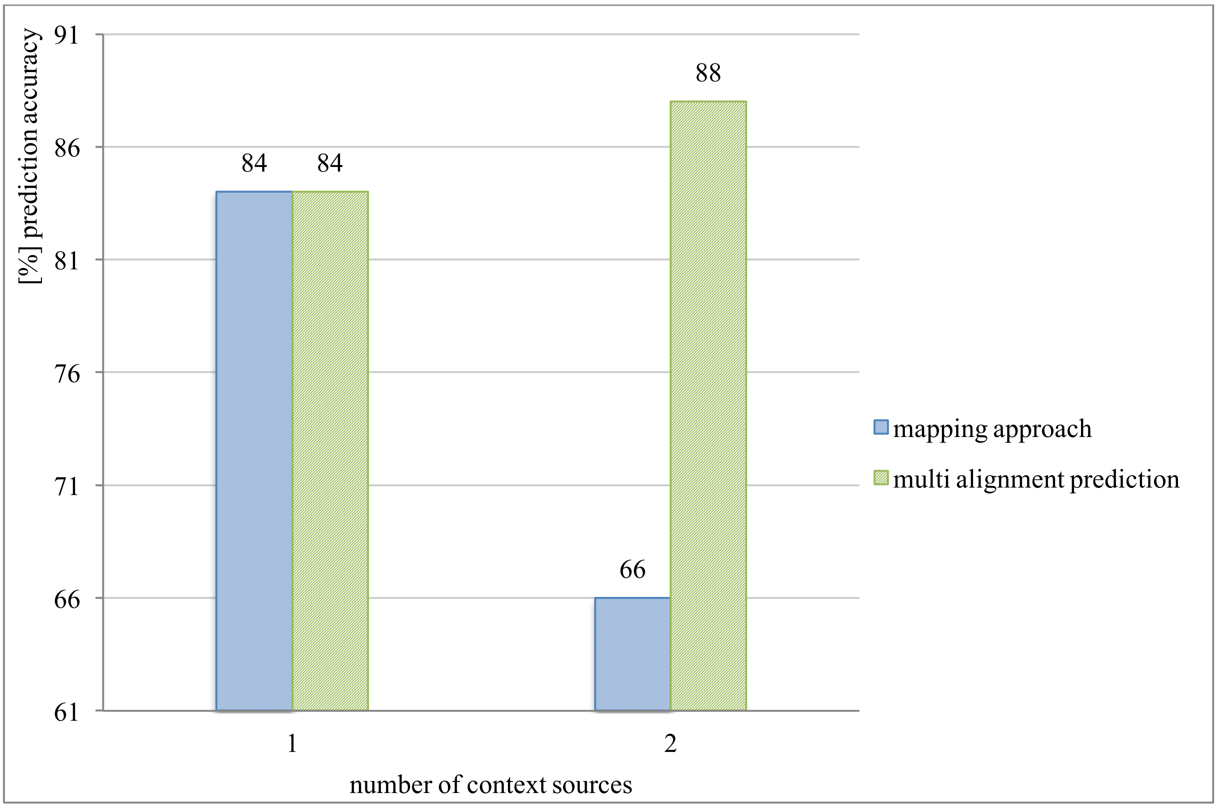


Figure 13: prediction accuracy over number of context sources for two different context prediction approaches, based on simulated context values

5.4.2 Real-world data results

Figure 14 shows the result of the experiment when real world data is used. The prediction accuracy is equal for both approaches when only one context source is used. When two context sources are used the prediction accuracy stays equal for the mapping approach. When using the multi alignment approach and two sources the prediction accuracy gains 7%.

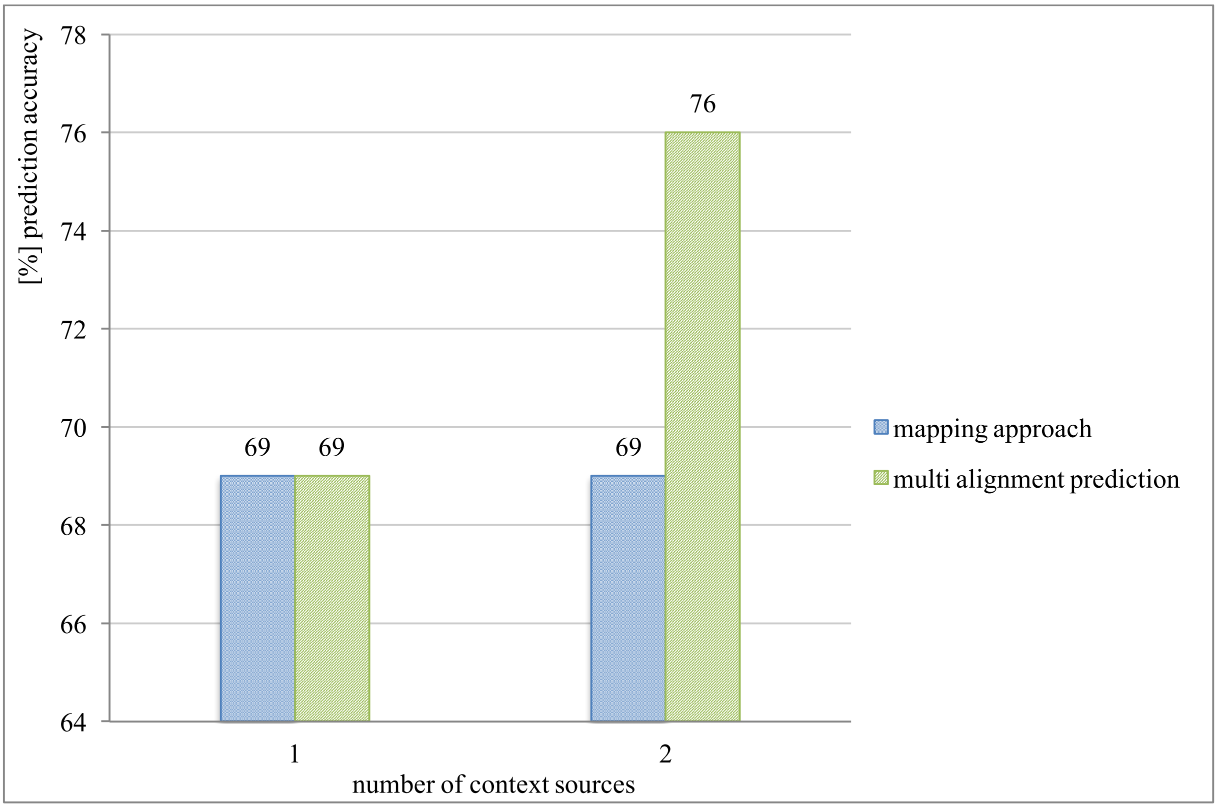


Figure 14: prediction accuracy over number of context sources for two different context prediction approaches, based on data from a real-world experiment

5.5 Discussion of the results

The prediction accuracy increased when multiple context sources and the multi alignment prediction approach were used. We expected an increase, as the added context source also adds information to the predictor. The following example may explain this in more detail. In Table 2 a sample from the real-world experiment data is shown. The column context source '*location*' shows a pattern where {'*work*', '*work*'} is once followed by '*tram*' and another time by '*take_a_walk*'. Without the *temperature* context, it is impossible to predict the next context value after the pattern {'*work*', '*work*'}. But when we consider the *temperature* context the ambiguities are resolved. The pattern {'*work*' & '*outside*', '*work*' & '*inside*'} is followed by '*tram*' and the pattern {'*work*' & '*inside*', '*work*' & '*outside*'} is followed by '*take_a_walk*'.

Table 2: short data sample from real world experiment

Row	timestamp	*context source 'location'*	*context source 'temperature'*

1	**1302266442761**	*work*	*outside*
2	**1302267151485**	*work*	*inside*
3	**1302267555217**	*tram*	*inside*
	...	…	…
4	**1302703828420**	*work*	*inside*
5	**1302704419010**	*work*	*outside*
6	**1302704508229**	*take_a_walk*	*outside*

The prediction accuracy did not increase when multiple context sources and the mapping approach were used, it either stayed equal or it even decreased. This was already proposed in the analytical comparison of the two approaches. The mapping approach should not be used for multiple context sources.

The prediction accuracy was equal for both investigated approaches when only one context source was used. Both approaches do exactly the same in this special case. The mapping maps each existing context value into new context values and starts a regular alignment based context prediction. The multi alignment prediction approach tries to predict a context value based on the history of the according context value. This is also a regular alignment based context prediction. This way the results for both approaches must be equal.

The prediction accuracy gain is higher with data from the real world than with data from the simulator. The gain of prediction accuracy depends on some factors like the correlation strength of the context sources, the number of faults in the histories and the predictability of the user influencing the sensors. We cannot tell how much each factor influenced our real-world experiment. However, the gain is promising for a real-world application of the multi alignment prediction approach.

The absolute values of the prediction accuracy are not relevant for the analysis of the experiment, but they show the potential of the context prediction. With the real-world data, we predicted the *location* context of the user which had 41 possible context values. With this high number the baseline is at 2.4% (the worst predictor should not drop below this line, as 2.4% prediction accuracy can be achieved when we just use a random generator). The achieved prediction accuracy of 76% is well above this value.

5.6 Conclusion

In this chapter, we have investigates the influence on multiple context sources on the context prediction accuracy as well as two different approaches utilizing multiple context sources. An experiment was performed on both, real world data and simulated data to measure the prediction accuracy.

The results of the experiment showed an increase of 4% – 7% of prediction accuracy when the alignment multi alignment prediction is used and a second context source is added. However, the results also demonstrated the prediction accuracy drops up to 18% when the mapping approach was used. Therefore, we can conclude the prediction accuracy can be increased using multiple context sources, when a prediction approach is used that is benefiting from the correlations in the context histories.

The context sources used in this chapter were chosen by us. However, a multitude of different context sources based on the increasing number of sensors in our environment is available to be used in context prediction. While the prediction may benefit from the combination of some context source, it may get irritated by the combination of others. The next chapter is presenting a method to determine beneficial context sources.

6. Selecting beneficial context sources to gain prediction accuracy

A certain combination of context sources may increase the prediction accuracy while another combination may decrease it. To choose a beneficial combination, the task is to answer: 'Is certain combination is useful?' The most intuitive way to answer the question may be to make some test predictions based on this certain combination of context sources and then compare the resulting prediction accuracy against the prediction accuracy without the combination or against other possible combinations. However, such a test approach can be time consuming depending on the selected prediction approach. Therefore, we propose a method to calculate the increase of the prediction accuracy beforehand, when multiple context sources are combined.

In this chapter, we describe a method to calculate the increase in prediction accuracy due to a combination of context sources, which is based on statistical properties of the context sources.

6.1 Selection approach

Our proposed selection approach can be described in three simple steps:

First: calculate the predictabilities. This has to be done for the context source you want to predict as also for each possible combination of context sources with the context source you want to predict.

Second: compare the predictabilities to find the combination with the highest predictability.

Third: choose the combination with the highest predictability for the prediction task. The next paragraphs explain how the predictability of multiple sources can be calculated.

The formula to calculate the predictability was explained in the State of the art section in section 3.3. The equation for the predictability is:

$$\mathcal{H}(\mathcal{X}) = -\Pi^{max} \log_2 \Pi^{max} - (1 - \Pi^{max}) \log_2 \frac{1 - \Pi^{max}}{N - 1} \tag{16}$$

$\mathcal{H}(\mathcal{X})$ is the entropy rate of a context source, N represents the number of different context values from a context source and Π^{max} is the predictability. The entropy rate $\mathcal{H}(\mathcal{X})$ has to be calculated, which can be very time consuming if the history is very long. The more practical way is the use of an Entropy estimator. This is also done by other authors mentioned in the state of the art section.

To estimate the entropy rate $\mathcal{H}(\mathcal{X})$ we used the Lempel-Ziv entropy estimator, which is

$$H^{est} = \left(\frac{1}{n} \sum_{i=1}^{n} \Lambda_i \right)^{-1} \ln n \tag{17}$$

n is the number of elements in the context history and Λ_i is the length of the shortest time series starting at position i in the history which does not appear before i.

Equation (17) only estimates the entropy rate of a single context source. Therefore, we have to add an additional step to the predictability calculation, to use it with multiple context sources.

When the entropy rate of a single context source is not estimated, but calculated, the conditional probabilities of the context values must be calculated (This was also shown in equation (2) in section 3.3). This means the probabilities of a certain context value given the condition a certain combination of context values have been the predecessor has to be calculated. However, as we want to consider multiple context sources we have to consider the condition of a certain combination of predecessor context values in the history of the first context source and the history of the additional context sources. The constrained combination of predecessor context values is achieved when the context sources are mapped into a single context source.

To always consider the appearance of certain combinations of predecessor context values of *all* context sources, we decided to add a prior mapping from multiple context sources into a new artificial context source. A mapping ensures that the combination of context values from multiple sources which occurred at a certain time is considered.

During the mapping, each combination of context values from the different context sources to be combined gets mapped into a unique context value. This mapping is illustrated in Figure 15. The exemplary context sources CS_1, CS_2 and CS_3 are chosen for a multi context source based prediction. To determine their predictability, they get mapped into a new artificial context source CS_*. The context values in the figure are symbols to remember the often non-ordinal character of context values.

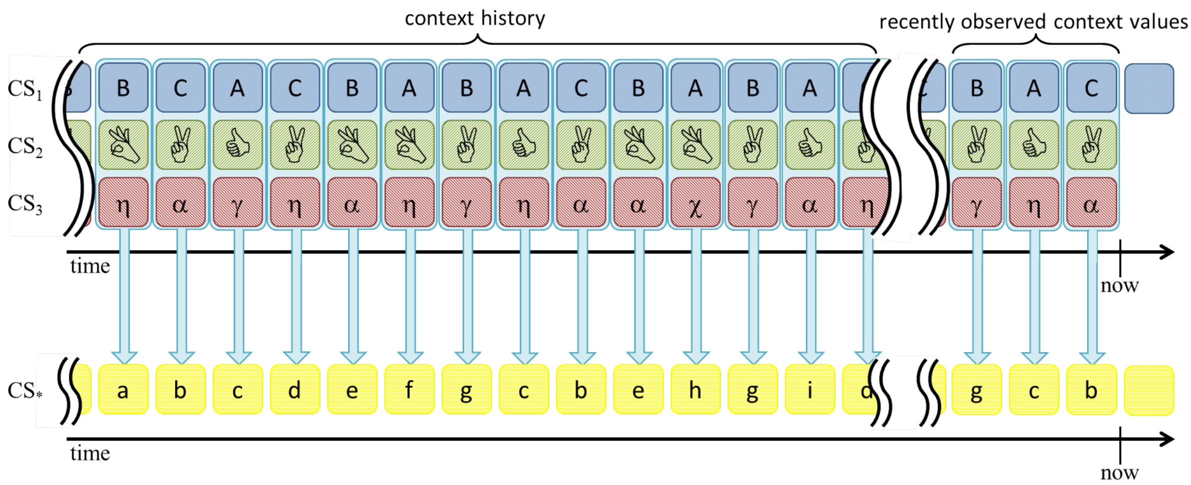


Figure 15: Mapping of context values into new context source CS_*

After the context sources have been mapped the entropy rate of the resulting artificial context source can been estimated with equation (17) just like from a single context source. Afterwards equation (16) can be used to determine the predictability of a certain combination of context sources. However, a solver has to be used to determine a numerical value for Π^{max} as it cannot be isolated in (16).

6.2 Experiment

We decided to add an experiment to verify our approach. The experiment with the data collection and the measurements are described in this section.

The experiment consisted of four steps. First we collected sensor data from two users and multiple sensors to derive context values in different context sources from the sensor data. Second we made a series of prediction accuracy tests with different combinations of the context sources. Third we calculated the predictabilities of different combination of the context sources. Finally, we were able to compare the predictability values with the prediction accuracy values of the different context source combinations. Accordance of the values is expected to verify our proposed approach.

6.2.1 Data collection

We collected sensor data correlated to the behaviour of two users. The users carried a Smartphone in their pocket over the test period of five hours. They randomly moved through a smart home according their daily tasks. About 6400 datasets were recorded. A new dataset was generated each time a context value changed.

A smartphone was used to record the contexts *activity*, *noise* and *floor level*. The *activity* context was derived by the approach described in [4], where the accelerometer of the smartphone is used to derive the context values '*sitting*', '*standing*' and '*walking*'. The microphone of the Smartphone was used to derive the *noise* context. The noise level was averaged over a short time spawn and then quantized into the five levels '*veryLoud*', '*loud*', '*average*', '*quiet*' and '*veryQuiet*'. The Smartphone barometer was used to derive the user's *floor level* context. An approach similar to [39] was implemented where the barometer value is quantized. As our smart home had two flor levels we derived the context values '*downstairs*' and '*upstairs*'.

The smart home used during the experiment was equipped with PIR motion sensors. They were used to record the user's *room* context. We experiment took place in six different rooms, so we simply numbered the context values from '*1*' to '*6*'.

6.2.2 Prediction accuracy test

After the context values were collected we made some measurements on the prediction accuracy. We utilized an alignment based approach able to use multiple context sources which is further described in section 5. The alignment approach uses a small series of recently observed context values to align them with the context history and predict future upcoming context values. Therefore, we split the collected context data into a context history part and into a test set of about 1/3 of the data. Then we successively took a small series of context values from the test set and feed it into the alignment predictor. We compared the predicted context value with the context value following the small series in the test set, which is what we expected to be predicted. If the two symbols were the same we evaluate the prediction as accurate, else as inaccurate prediction. After we repeated this over the whole test set we calculated the average of accurate predictions, the prediction accuracy.

We decided to predict the room number the user will be in. Thus we made the prediction accuracy test with each possible combination of the other context sources with the room context. We also did this for both users.

6.2.3 Predictability calculation

The last step of the experiment was to calculate the predictabilities. First we mapped each possible combination of context sources with the room context. Then we calculated the entropy rates for each mapped context source. Afterwards we used a solver to calculate the predictability for each combination. We did this for both users.

6.3 Results

In this section the results are presented. Table 3 shows the prediction accuracy and the predictability measurements. The results are separated by user and also by the context source used for the measurement. All numbers are percentages.

Table 3: Results of the predictability and prediction accuracy comparison

	User 1		*User 2*	
	predictability	*prediction accuracy*	*predictability*	*prediction accuracy*
Room	72%	27%	80%	41%
room & noise	73%	29%	70%	36%
room & activety	76%	27%	79%	41%
room & floor level	72%	27%	80%	41%
room & noise & activety	74%	17%	74%	26%
room & noise & floor level	73%	29%	70%	36%
room & activety & floor level	76%	27%	79%	41%
room & noise & activety & floor level	74%	17%	74%	26%

Figure 2 and Figure 3 show the results in graphs. The y axis is always the percentage of prediction accuracy and predictability. The graphs help to identify the trends of prediction accuracy and predictability. An interpretation of the results is given in the next subsection.

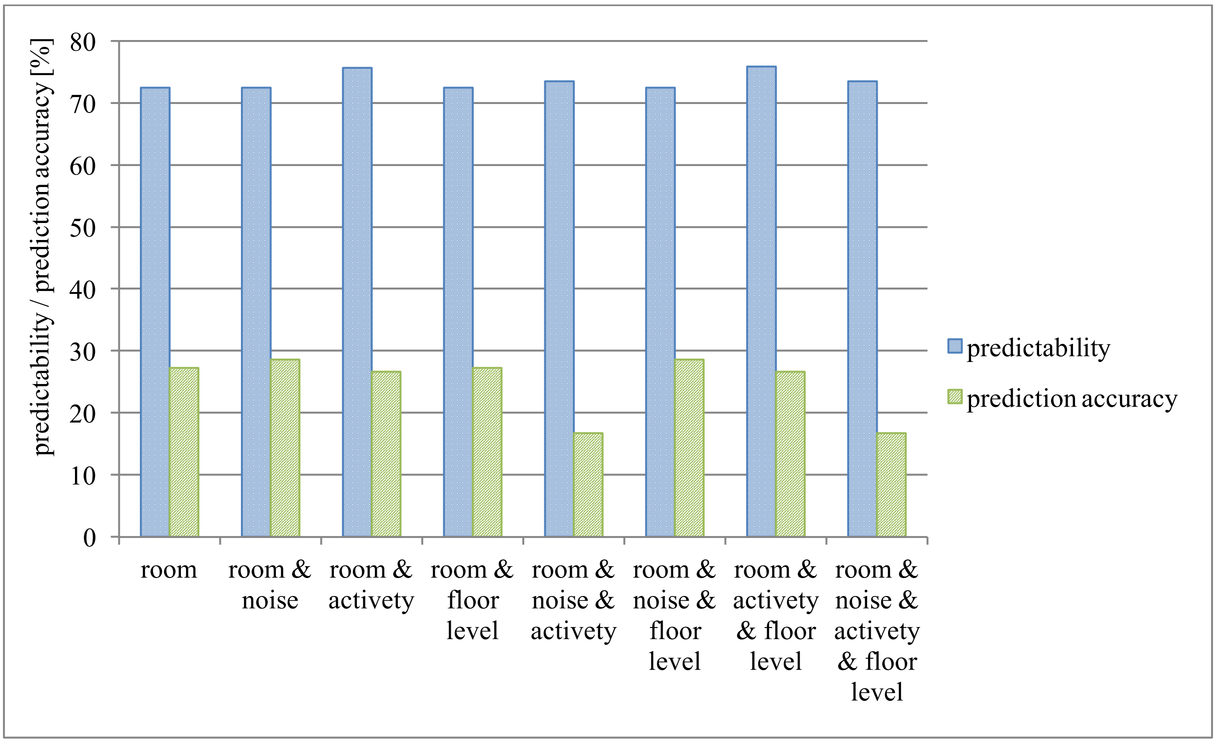


Figure 16: Results based on data from user 1, comparison of tendency between predictability and prediction accuracy.

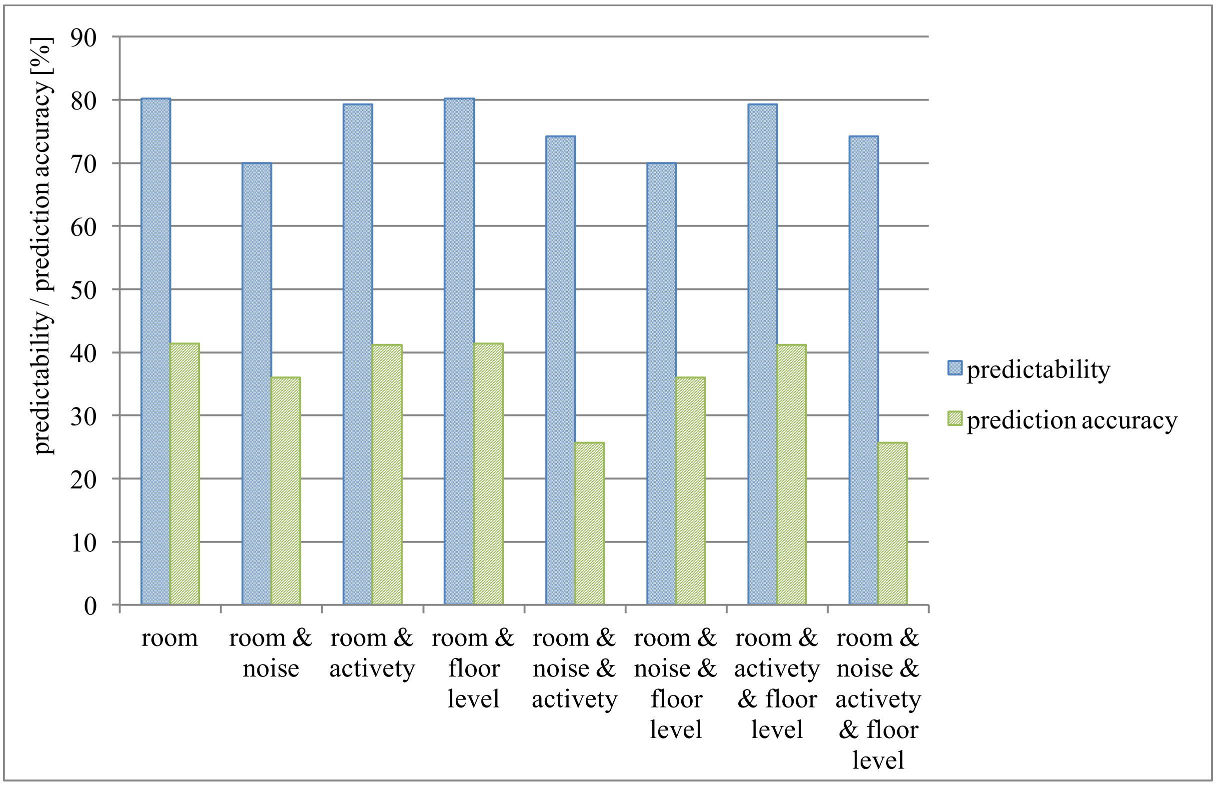


Figure 17: Results based on data from user 2, comparison of tendency between predictability and prediction accuracy.

6.4 Discussion of the results

The results are interesting in several aspects: first we want to know if the prediction accuracy increases when the predictability also increases for a certain combination of context sources. Second we will discuss why the absolute values of the predictability and prediction accuracy differ so much. And third it is also interesting to compare the difference between user 1 and user 2 for the selection of beneficial context sources.

6.4.1 Dependency on the combination

In most cases the prediction accuracy rises when the predictability also rises. This result verifies our approach. The predictability can be used to check beforehand whether a certain selection of context sources will increase prediction accuracy or not. However, in some cases (for user 1 every time the activity was included) the predictability and the prediction accuracy did not correlate so well. Most likely this happened due to the only a fife hour context survey. The predictability is a statistical value which. Therefore, the accuracy of the statistical properties, including the predictability, is relying on the size of the utilized dataset. A too short dataset can decrease the accuracy of the statistical properties. Nevertheless, the predictability and prediction accuracy values de-correlate not much.

6.4.2 Difference between predictability and prediction accuracy

Although the absolute values of the predictability and the prediction accuracy do not matter for the verification of the proposed selection approach, it might be interesting to discuss the difference between them.

An optimal prediction algorithm should be able to reach the predictability with the prediction accuracy. The prediction approach we chose performs only half as well as the optimal predictor. For the design of a prediction system for daily life this can be an indicator to change the prediction algorithm in order to get a better prediction performance. For our test, it was not necessary as we were only interested in the relative values.

6.4.3 Influence of the user on the selection of context sources

For user 1 the combination of the room and noise context was gaining the prediction accuracy. The same combination did decrease the prediction accuracy for user 2. This is because the patterns in the context values are related to the specific user behaviour. Therefore, an optimal combination must be selected for each user individually.

6.5 Conclusion

In this chapter, we presented a context source selection approach. It can be used to select beneficial context sources for context prediction, in order to gain the prediction accuracy. Therefore, we utilized the predictability derived by song [23]. We made an extension to the predictability formula to use it for multiple context sources. Afterwards we made an experiment, where we collected sensor data from different smartphone sensors and different users, derived context values and performed prediction accuracy measurements with different combinations of context sources. We also calculated the predictabilities on different combinations of context sources. The accordance of the predictabilities with the prediction accuracies verified our approach. When the predictability increased due to a beneficial combination of context sources also the prediction accuracy increased and vice versa. Therefore, our presented approach can be used to make a prior selection of context sources when some context values are already collected.

7. The effect of prediction stability against disturbances of sensors[3]

The prediction accuracy of a certain context prediction algorithm does not only rely on the algorithm itself but also on the utilized context history. Faulty data in the context history may reduce the prediction accuracy. The amount of faulty data in the history can vary, depending on the source of the fault. One example for a faulty context generation is a disturbed sensor. From time to time a certain sensor may pick up some electrical noise and therefore cause faulty data. Another example could be a faulty reasoning in the context source. A context source usually reasons context values with an algorithm trained on previous sensor data. If unknown sensor data occurs, a faulty context may be derived by the context source. However, a combination of multiple context sources can make a prediction system more stable against disturbances.

A combination of multiple context sources can use redundancies to increase the stability against disturbances. While some disturbances cause faulty data in some context sources, others context sources may not be influenced by the same disturbances. When a prediction at a certain time is only based on one single context source which then gets disturbed, the prediction is likely to be inaccurate. But when the same prediction also considers other context sources, the fault can often be compensated by utilizing redundant information from the non-disturbed context sources.

Redundant information is generated when multiple sensors are related to the behaviour of one user. This can be better understood by the help of Figure 2. The Sensors $S_1 \ldots S_m$ get influenced by the user's behaviour, represented through the cloud with the different things the user might do. The context sources $CS_1 \ldots CS_n$ derive different context values from the sensor data. If the user does a certain activity, most likely several sensors $S_1 \ldots S_m$ will be influenced at the same time. The information generated by CS_1 will then for example have redundancies to CS_2.

[3] Parts of this chapter have been published in [37].

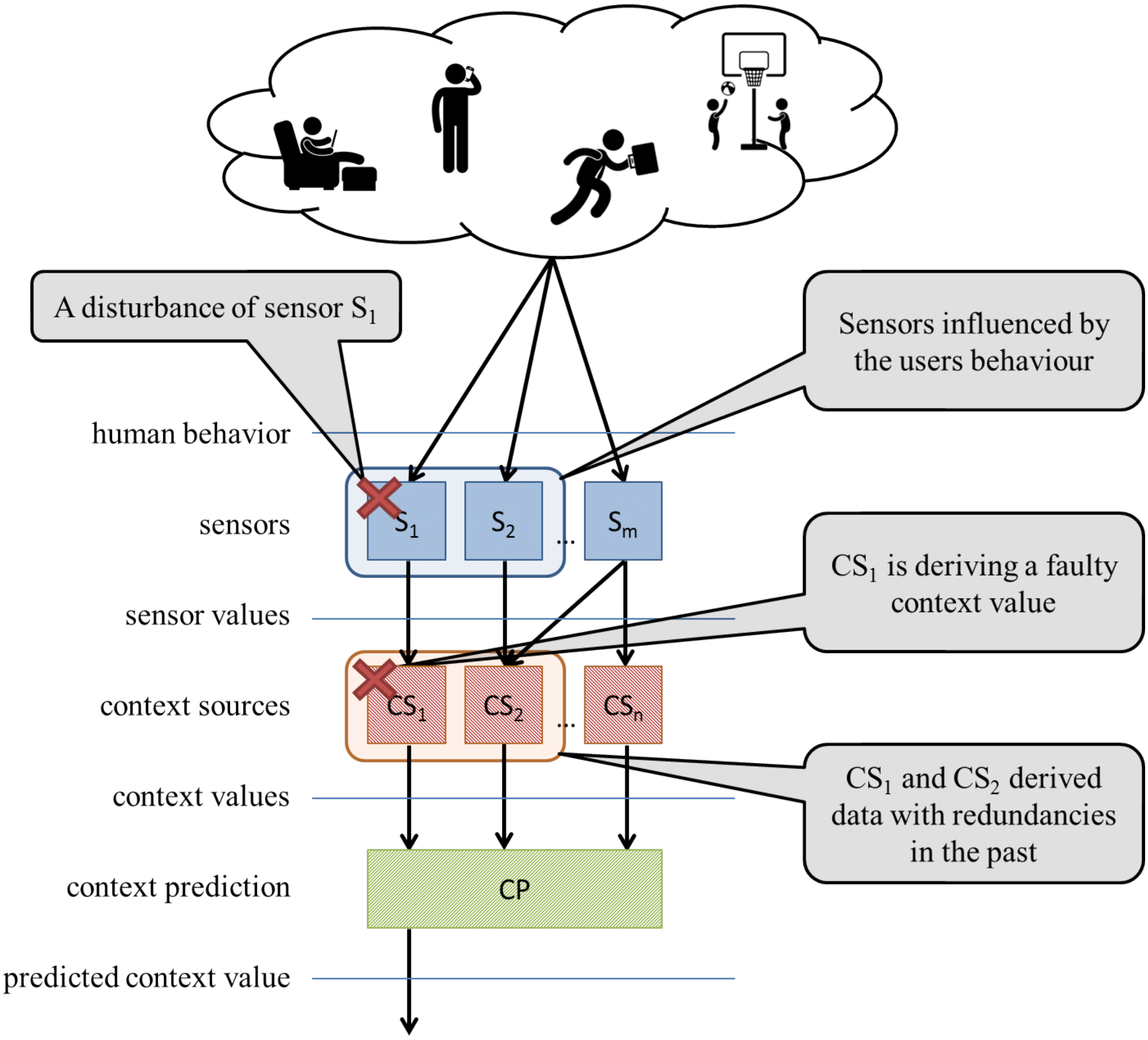


Figure 18: Sensor S_1 and context source CS_1 get disturbed

Redundancies can be used to fix faulty context data. The context prediction algorithm CP predicts the upcoming context value for context source CS_1, based on multiple context sources. The prediction algorithm is not only based on the context source CS_1 but using all context sources $CS_1 \ldots CS_n$. Figure 18 shows an example where the sensor S_1 gets disturbed at a certain time. The context source CS_1 might then derive faulty context data. This is an information loss in CS_1. But the context predictor CP has access to redundant information from $C_2 \ldots C_n$. This can be used to compensate the information loss of the disturbed C_1.

In this chapter, we present an investigation on the effect of stability against disturbances when multiple context sources are combined for context prediction. First the methods of the investigation are explained. Then the results are presented and afterwards discussed.

7.1 Methodology of the experiment

We designed an experiment to measure the effect of stability against disturbances, when multiple context sources are used. The experiment was done with two datasets of context histories, the first one was generated by a simulator and the second one was collected by a real-world experiment. Afterwards artificial faults were introduced into the context histories. The number of faults was increase in 10% steps from zero to 50%. The faulty context histories where then used in a multiple context prediction algorithm and the effect of the faults on the prediction accuracy was investigated.

The process of the artificial faults introduction, the simulation and the real-world experiment are further described in this subsection. Also, the decisions on why a real world experiment was conducted and the measurement of the prediction accuracy are explained.

7.1.1 Artificially introduced faults

To investigate the effect of multiple context sources on the stability of the prediction accuracy against disturbances, we decided to artificially introduce disturbances. In contrast to naturally occurring disturbances, artificial introduced disturbances can be easily controlled. Disturbances which occur naturally can sometimes be quantified but usually not influenced.

Two randomly selected context values inside a context history were exchanged to introduce a disturbance. This corresponds to a reasoning error in a context source. An alternative way to disturb the data would be to override randomly selected context values in the history with symbols like ‘not available’ or ‘unknown’. However, such new symbols can easily be identified by the prediction algorithm, but the exchange of two context values cannot be identified. Hence our chosen exchange approach results in a more difficult task to the prediction algorithm.

We used a self-designed algorithm to exchange the context values in the context history. First the desired percentage of disturbance was set. Then the algorithm exchanged a pair of context values. Next the resulting amount of disturbances was calculated by comparing the original context history with the disturbed one. The calculation had to be done as a repetitive exchange at a certain position in the context history could easily restore the original context value at this position in the context history. The process was repeated until the desired amount of disturbances was reached.

7.1.2 Simulated Data

We decided to use both, data from a simulation and data from a real-world experiment. The simulation did not include disturbances in contrast to the real-world experiment data, where always some disturbances are inherently included. We also cannot tell the quantity of the inherent disturbance in the real-world data. Hence the simulation data is more useful for an investigation on the effect of disturbances. Nevertheless, the real-world data is useful to ensure the experiment has no bias from the simulator.

We used the data from the simulation described in 5.3.1 which is based on SIAFU, a context simulator. Fourteen different agents were recorded and over 94000 datasets for each agent. We made the experiment with multiple agents, but as the results showed the same trends and to keep the results section focused, we only show the results of one agent.

We chose to use the following four context sources: *activity*, *noise level*, *office area* and *Wi-Fi reception* from the simulator. With this selection, we have chosen context sources similarities to the context sources recorded in real world experiment. The real-world experiment is described in the next subsection.

7.1.3 Experimental Data

The real-world experiment data was collected with both, a Smartphone and a smart home. A person was carrying a Samsung Galaxy III Smartphone in his pocket while he was moving through several rooms of a smart home. The rooms were distributed over two floor levels.

The Smartphone collected the three contexts *user activity*, *floor level* and *noise level*. The *user activity* was derived with the approach shown in [4] where a Smartphone accelerometer is used to derive the human activities '*sitting*', '*standing*' and '*walking*'. The *floor level* was derived by an approach similar to [39] where a Smartphone barometer measures the air pressure. With each floor level the pressure changed about 0.5hPa. A two-level quantization was deriving '*upper floor*' or '*lower floor*'. The *noise* *level* was derived by averaging the microphone input of the Smartphone and quantising it afterwards to fife levels from '*low*', '*medlow*', '*med*', '*medhigh*' to '*high*'.

The smart home was used to collect the *location* context. Eight rooms of the smart home were used for the experiment, all equipped with presence detectors. The presence of the person moving through the house was recoded and simply represented by eight room numbers. Some of the rooms had a direct pass way between each other; some were connected through a hallway, which was also one of the monitored rooms.

We recorded the contexts of two persons in over 3800 datasets from each. To verify our experiment, we did the accuracy measurement with the dataset of both persons, but as the results looked similar the results section only contains the results of one person.

7.1.4 Measurement of prediction accuracy

The measurement was done in two steps. First we made a series of predictions. The predictions were done with two different prediction algorithms, one which uses only one context source and one which uses multiple context sources. And the predictions were done with all disturbed context histories, reaching from 0% - 50% disturbances. The second step was to evaluate how accurate each approach had predicted upcoming context values, even under the influence of different amounts of disturbances. With this two-step procedure, we can compare the influence of disturbances on a multi context prediction approach with the influence of disturbances on a single context prediction approach.

We used an alignment based predictor, described in chapter 5.1.1for the prediction. We used the same prediction algorithm in both cases, only once with the extension described in chapter 5.1.3 and once without the extension. This enables us to compare the results, without considering different prediction accuracies caused by different algorithms. When using the simulated data, we predicted the future activity context and when using the real word data, we predicted the future location context.

Before we run the predictions, we split each dataset from the simulation data and real world data into two parts. One part of each dataset served as context history. This part was about two third of the length of the original dataset. The shorter part of about one third served as a test set. The test set was used to run the prediction accuracy measurement.

We measured the accuracy of a prediction by comparing the predicted context value with an expected context value. The alignment based context predictor uses a short sequence of about four or five recently occurred context values to aligning the sequence with the context history. We took the short sequence from our test set and started the predictor. The context value following out taken sequence in the test set is what we expected to be predicted. The predicted context value was compared to the expected context value and either be rated as accurate or inaccurate. This procedure was repeated over the whole test set. Afterwards the average percentage of accurate predictions was calculated. We call this average the prediction accuracy.

When we made the measurements with the disturbed histories, we took the short sequence of context values from a test set that was also disturbed the same way the history was. It was necessary to use a disturbed test set when using a disturbed history. Disturbances will not only affect context values stored in the context history but also the last view context values that were derived by a context source. In contrast to this we took the expected context value, the context value following the position of the sequence in the test set, from the undisturbed test set. This was necessary as we defined the expected value as the truly future context value. Using a disturbed context value here would be wrong as the future upcoming context value cannot be disturbed and the prediction accuracy measurement would be faulty.

7.2 Results of the experiment

In this section the results of the experiment are presented. There are two sections, one for the results based on the simulation data and on or the results based on the real-world data. First the results are given in a table. Afterwards the graph is shown to visualize the trend.

7.2.1 Results based on measurements with simulated context data

Table 4 shows the result of the experiment on the influence of disturbed context values on different context prediction approaches. A context prediction approach that utilizes only one context source massively decreases prediction accuracy, as the amount of disturbances increase. The amount of disturbances also has a decreasing effect on the prediction approach utilizing four context sources, but only in a much lesser extent. The huge difference in the impact of the disturbance can be even better seen in Figure 19, where the green bars tend much faster against zero then the blue bars. The absolute prediction accuracy plays only a minor role in this chapter, as we focus on the changes caused by disturbances.

Table 4: prediction accuracy with context history disturbance based on context simulator data

context history disturbance	*0%*	*10%*	*20%*	*30%*	*40%*	*50%*
one context source	84%	71%	53%	35%	24%	11%
four context sources	88%	88%	82%	77%	73%	71%

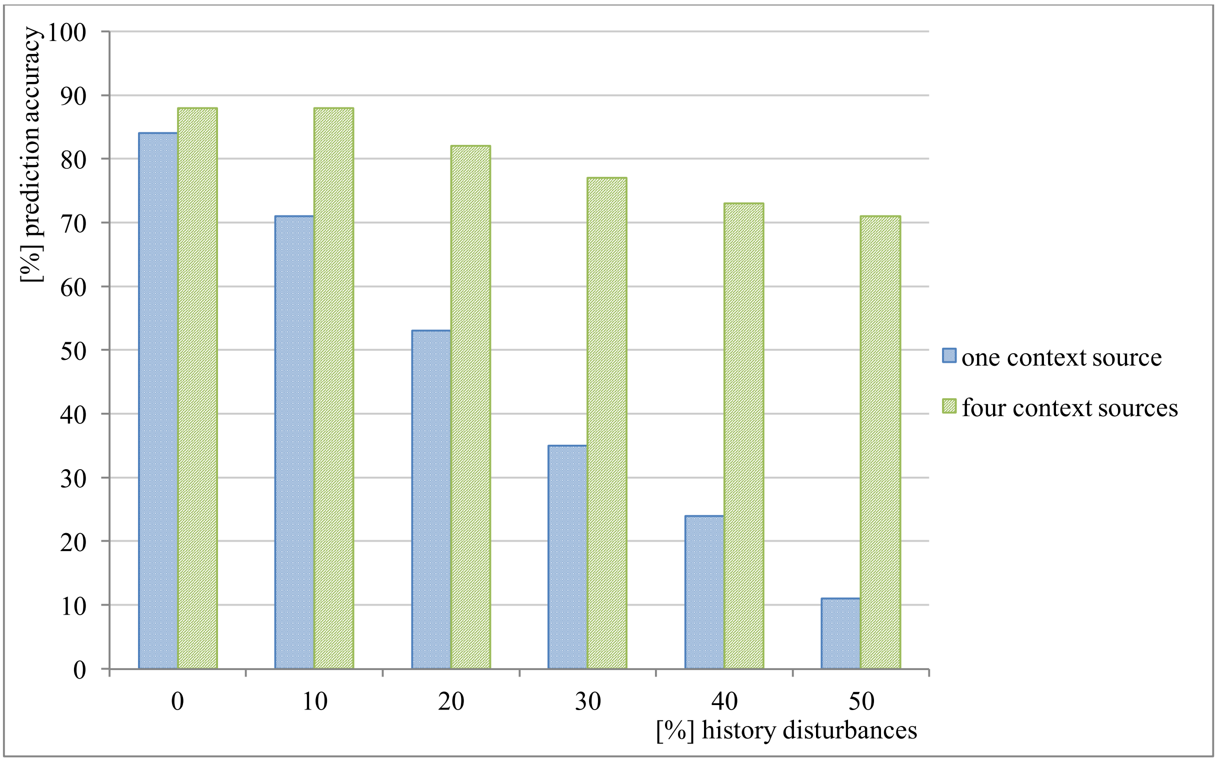


Figure 19: Comparison of the influence of disturbances when one ore multiple context sources are used, based on disturbed simulation data

7.2.2 Results based on measurements with real-world data

The results from the experiment based on real world data are similar to the results based on the simulation data. Although the impact of the disturbances is smaller, the prediction accuracy of the approach utilizing only one context source still decreases strongly. At the same time the prediction accuracy of the approach utilizing four context sources stays constant.

Table 5: prediction accuracy with context history disturbance based on real world data

context history disturbance	*0%*	*10%*	*20%*	*30%*	*40%*	*50%*
one context source	81%	80%	71%	67%	51%	46%
four context sources	82%	82%	82%	82%	82%	82%

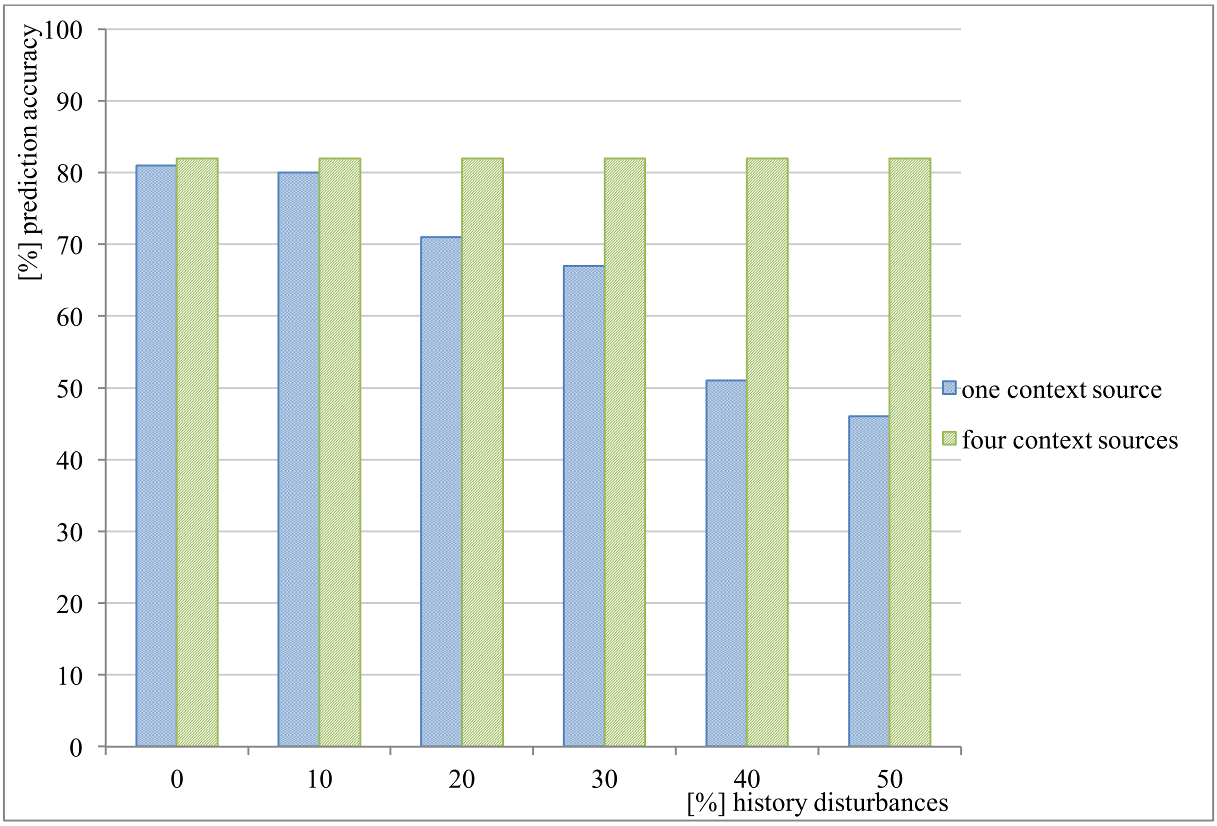


Figure 20: Comparison of the influence of disturbances when one ore multiple context sources are used, based on real world data

7.3 Discussion of the results

In this section the results are discussed. First we are going to discuss the impact of disturbances on a prediction approach utilizing only one context source. Then we will discuss the impact on a prediction approach utilizing multiple context sources. Finally, we are going to compare the results based on the real-world data with the one based on simulated context data.

7.3.1 Impact of disturbances on the multi context source approach

The prediction accuracy of our multi context source approach stayed at a high level, when disturbances were added. This is in accordance with our preliminary consideration which is: a multi context source approach can compensate faults by using redundancies added using multiple sources. However, there are also redundancies in the context values derived by a single context source. This happens for example when a user does the same thing twice. But when the last few generated context values get disturbed, it gets difficult for a prediction algorithm to find an accurate prediction. The algorithm get irritated as the actual situation differs from what the context source delivers to the algorithm. From not knowing the actual context value properly it is even with redundancies in the context history difficult to predict future context values. This is where the multi context source approach gains most of it disturbance stability. Even if the actual situation differs from what one context source delivers to the algorithm, the other context sources may still deliver a proper description of the actual situation.

The low influence of disturbances on the multi context source approach is remarkable. In the experiment with the real-world data the accuracy dropped only about 17% and in the experiment with the simulated context data the accuracy stayed constant when the disturbance level was at 50%. This means half of the information in the context history was wrong or faulty and the prediction system was still useable. These numbers can of course not be generalized as the actual stability always depends on factors like the correlation strength of the sensors and the user's behaviour. But it can clearly be seen: a prediction approach based on multiple context sources adds stability against disturbances.

7.3.2 Impact of disturbances on the singe context source approach

The impact of the disturbances was much larger on a single context source approach. The single source approach had only minor possibilities to compensate the information loss caused by the disturbances. In the simulation data experiment the prediction accuracy dropped to about 11% prediction accuracy. This is close to the bottom line of prediction accuracy, when predicting a context source with eight possible symbols. Even a simple random generator would attempt a prediction accuracy of 12.5% by simply guessing each possible symbol with the same probability. This means the single context source approach was not useable anymore at a disturbance rate of 50%.

7.3.3 Differences between simulation and real world data

The results from the experiments based on real world data and on simulated context data showed similar trends. The disturbance had a bigger impact on the single source based approach and disturbances decreased the prediction accuracy. However, there was also a difference in the amount the prediction accuracy was decreased by the disturbances. This can have multiple causes. First, the behaviour of the users also influences the amount of redundancies in the context history. A very repetitive user will cause a lot of redundancies in the context history and thus disturbances will not influence the prediction accuracy this much. Second, the sensors were differently strong correlated to the user's behaviour. The *noise level* context in the context simulator fore example was also influenced by the other agents and not only by the user himself, whereas the *noise level* context in the real-world data was mainly influenced by the user himself, as it was pretty quiet during the data collection. Third cause can be the dependencies in the order of the context values. The *location* context in the real-world experiment for example was collected by monitoring the room the user is in. Some rooms can only be reached by passing other rooms like the hallway or the stairwell. Such arrangements generated conditional information in the history, which was also used by the prediction algorithm. The amount of such conditional information is depending on the user's behaviour and the physical constrains of the environment, which was not the same in both data collections.

7.4 Conclusion

When engineers design a communication system they will usually add redundancies to the transmitted information to make the communication robust against disturbances. And usually the can decide how much redundancies the want to add. When we design a context aware system with multiple sensors and multiple context sources, we can usually not decide how much redundancies are generated by the different context sources. Nevertheless, we can also use the given redundancies to make a context prediction system more stable against disturbances.

In this chapter, we showed that a context prediction system can be more stable against disturbances by utilizing multiple context sources. We investigated this by an experiment where we measured the prediction accuracy in relation to artificially introduced disturbances. The results showed it is possible to keep the prediction accuracy high when using multiple context sources for prediction, whereas the prediction accuracy will significantly drop when only one context source is use and disturbances are added.

8. Energy usage on multiple sensors of modern Smartphones[4]

Prediction future context values, using multiple context sources and increasing the prediction accuracy needs of course devices to be carried out on. One very suitable device is the Smartphone. It is usually in close proximity to a user a therefore also influenced by his behaviour. Smartphones are equipped with multiple sensors like an accelerometer, gyroscope, compass, GPS, barometer, light sensor, proximity sensor and some even have a pulse sensor. This variety of sensors enables Smartphones to collect sensor data. Current Smartphones also have powerful processors with multiple cores and with more than one GHz. This enables them to process the collected sensor data and derive context values, such as the mode of transportation [53][54][55] or the user activity like "sitting", "standing" or "walking" [4]. The processing power can also be used to predict future upcoming context values. And with LTE or 3G current Smartphones have a fast connectivity, which can be used to trigger actions, based on current or predicted context values. On the other hand, using a Smartphone for context acquisition and prediction also increases the energy usage of a Smartphone.

The energy usage of applications for a Smartphone, including context acquisition and prediction, is important. The energy amount of a Smartphone is strictly limited due to the battery life. An application which increases the energy consumption will decrease the battery runtime. Context aware applications are potentially energy-intensive. They are usually running in the background gathering sensor data. This is why "even a small power requirement has the potential to impact the device more heavily than other power hungry but short-lived programs" [5]. Kanhere pointed out how critical this problem is when he observed that, „people will stop participating if such applications use up their phone battery"[6].

To understand whether a modern multiple sensors equipped Smartphones is useable for daily sensing, in terms of energy consumption, we conducted an experiment. We investigate the energy consumption of different Smartphone sensors in two ways: First by measuring with a volt meter and an ammeter and second by the Smartphone internal

---

[4] Parts of this chapter have been published in [38].

energy API. This approach enabled us not only to gather the needed information but also to compare the Smartphone energy API with actual measured values. We did our experiment exemplary with a Samsung Galaxy II (SGII), Samsung Galaxy 3 (SGIII) and an iPhone 4.

In this chapter first the methodology is presented. Afterwards the results are presented and discussed. This also concludes the answer whether an average Smartphones can be used for a daily sensing and context survey or not. Finally, the conclusion is drawn.

8.1 Methodology

We measured the energy consumption of different Smartphone sensors and the influence of a certain sensor usage on the battery runtime. This enables us to conclude whether a daily sensing and context survey is feasible with an average Smartphone in terms of battery runtime. From this measurement, we can also conclude how much a certain sensor and therefore a context derived from this sensor costs in terms of energy or battery runtime.

8.1.1 Energy measurement

The sensors were built in the Smartphone during the measurement. This approach was chosen to make the energy measurements simple and practical. One alternative approach would be to unsolder the sensors and measure their energy consumption on the workbench. Unfortunately, the high integration in modern Smartphones makes it impossible to get a single sensor. Multiple sensors are usually combined in single chip. The other alternative approach of using the datasheets to simply lookup the energy consumption was also not possible. The sensor chips used in Smartphones are customized to the Smartphone manufactures constrains and they usually to not publish datasheets of such customisations. Our approach of testing the sensors in system is also much closer to the aimed scenario of using the sensors inside the Smartphone in daily life.

The energy consumption was measured in two different ways. First we started with measuring the energy consumption of different Smartphone sensors by utilizing the Smartphone internally provided energy usage API. But to validate our measurements we decided to repeat the same measurement with a volt meter and an ammeter connected to the Smartphone battery. This approach also enables us to compare the differently made measurements and examine the API precision.

The API based measurement was made by utilizing the Smartphone internal energy APIs. As the APIs provide different values, depending whether an Android or iOS based phone is used, we had to use two different approaches. In Android based devices the energy consumption is provided directly. Therefore, we wrote an application which logs this energy consumption. In iOS based devices the battery status is provided. We wrote an application to log the battery status and calculated the energy consumption afterwards.

For the meter based measurements we used two PeakTech 3415 USB digital data logger multimeters. They have an accuracy of ±1.5% in current range and ±0.5% in voltage range. The battery of a Smartphone was removed during measurement and connected to a special adapter. This enabled the interconnection of the meters between the Smartphones and their batteries. The schematic setup and a picture of an actual measurement are shown in Figure 21. The multimeters were also connected to a PC which stored the measured values into a file with a sample rate of 2 Hz.

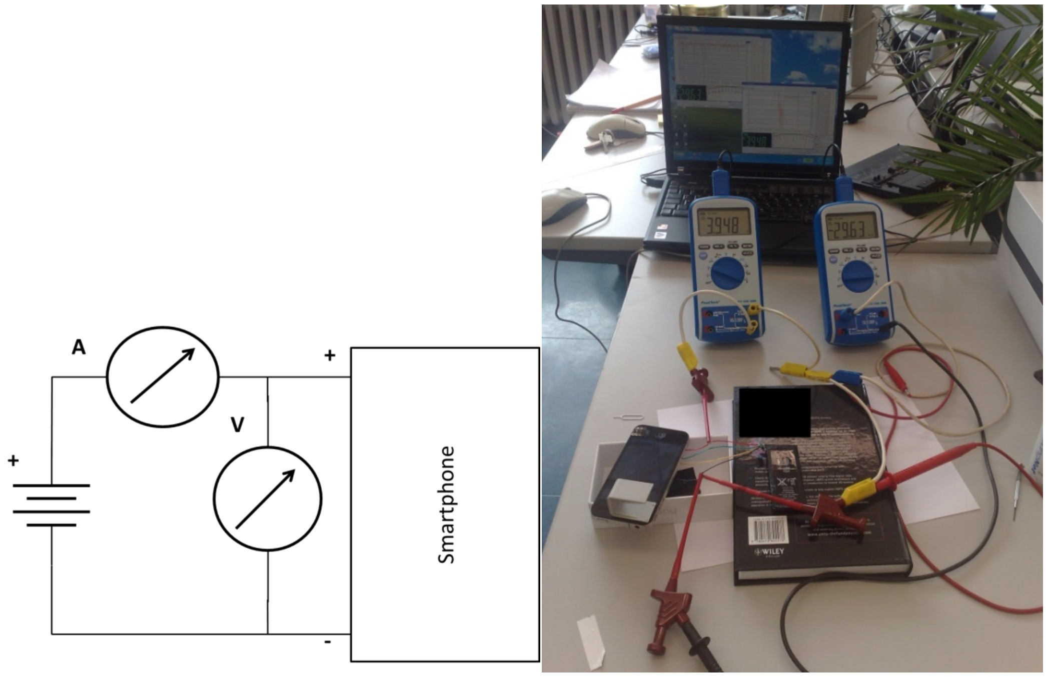


Figure 21: Schematic setup of the measurement(left) and measurement setup (right)

8.1.2 Investigated Smartphones and sensors

We used four different Smartphones to measure the energy consumption: two different models of the Samsung Galaxy II (GT-I9100 and GT-I9100G), a Samsung Galaxy III (GT-I9300) and an iPhone 4 (MC603DN). We did our first series of measurements with the Samsung Galaxy II and the iPhone. But the results varied widely so we decided to add another hardware revision of the Samsung Galaxy II and a Samsung Galaxy III. As it turned out the additional measurements with the other Samsung Galaxy versions were much closer to one of the first measurements. The large diversity between our first two measurements is discussed later in the discussion section.

The investigated iPhone showed some restrictions for the measurement. In the Samsung phones the sensors can be switched on and off separately. That is not possible for the iPhone. If the iPhone enters standby mode some processes are stopped, like sensing with the accelerometer. Therefore, we avoided the standby mode by deactivating the idle time of the iPhone. This lead to a problem concerning the display. The display is automatically switched off either by the idle timer or by the proximity sensor. To keep the measurement realistically, the display had to be turned off, as most of the time in a daily use of a Smartphone the display is turned off. Thus we decided to switch the display off by covering the proximity sensor. Therefore, the proximity sensor had to be turned on during all measurements. Another limitation of the iPhone is the accelerometer cannot be turned off, even when the rotation lock is activated. And a last limitation of the iPhone was a more practical problem. The back cover of the phone had to stay unattached during our measurement setup, to have access to the accumulator. However, with a removed back cover it was not possible to activate cellular connection on the iPhone.

The sensors whose energy consumption should be measured had to be available on all devices, to compare the measurements afterwards. Therefore, we chose four sensors: the accelerometer, the proximity sensor, the gyroscope and the compass. We also decided to do add a measurement with an activated 3G connection and activated sensors. According the limitations described in the last paragraph, we did five different combinations of sensors and activated 3G:

1. only the proximity and the accelerometer activated,

2. the proximity sensor, accelerometer and gyroscope,

3. the proximity sensor, accelerometer and compass,

4. the proximity sensor, accelerometer, gyroscope and compass,

5. the proximity sensor, accelerometer, gyroscope and compass with an activated 3G connection.

8.1.3 Measurement procedure

All measurements were repeated three times and the average was taken afterwards. Each measurement was done with a certain Smartphone model and a certain sensor combination. The procedure of a single measurement was: fully charge the Smartphone and disconnect the phone from the charger, connect the measurement setup between the phone and the battery, activate a certain combination of sensors and then measure the energy consumption.

To measure the energy consumption of a sensor as accurate as possible while it is Smartphone built in, side effects have to be reduced as much as possible. Before the measurement was started the phone was rebooted. Wi-Fi and Bluetooth were deactivated. The 3G connection was only activated in measurement number five. Only our measurement application was running. All other applications were closed with the multitasking toolbar or the task manager.

8.2 Results of energy measurements

In this section the results of the measurements are shown. Each Smartphone model is presented in a single table. Each table contains the measurements one till five and is made of two sections, one for the results of the API based measurements and one for the results of the voltmeter and ammeter measurements. The 'Average P' column contains the measured values. The 'ΔP' column contains the calculated difference caused by a single sensor or combination.

Table 6: Measured consumptions of iPhone 4 sensors

	Power consumption measured using software API	Power consumption measured using volt meter and ammeter

Sensor	*Average P*	ΔP	*Average P*	ΔP
Accelerometer, Proximity	102.90mW	---	121.99mW	--
Accelerometer, Proximity, Gyroscope	153.13mW	50.32mW	175.59mW	53.60mW
Accelerometer, Proximity, Compass	163.31mW	60.41mW	188.40mW	66.41mW
Accelerometer, Proximity, Gyroscope, Compass	204.28mW	101.38mW	212.37mW	90.38mW
Accelerometer, Proximity, Gyroscope, Compass, 3G	222.91mW	120.01mW	--	--

Table 7: Measured consumptions of Samsung Galaxy II GT I9100 sensors

	Power consumption measured using software API		Power consumption measured using volt meter and ammeter	
Sensor	*Average P*	ΔP	*Average P*	ΔP
Accelerometer, Proximity	436.24mW	--	514.55mW	--
Accelerometer, Proximity, Gyroscope	895.20mW	458.96mW	996.49mW	481.94mW
Accelerometer, Proximity, Compass	976.8mW	540.56mW	1102.13mW	587.58mW
Accelerometer, Proximity, Gyroscope, Compass	1294.34mW	858.10mW	1510.41mW	995.86mW
Accelerometer, Proximity, Gyroscope, Compass, 3G	1365.3mW	929.06mW	1686.85mW	1172.3mW

Table 8: Measured consumptions of Samsung Galaxy II GT I9100G sensors

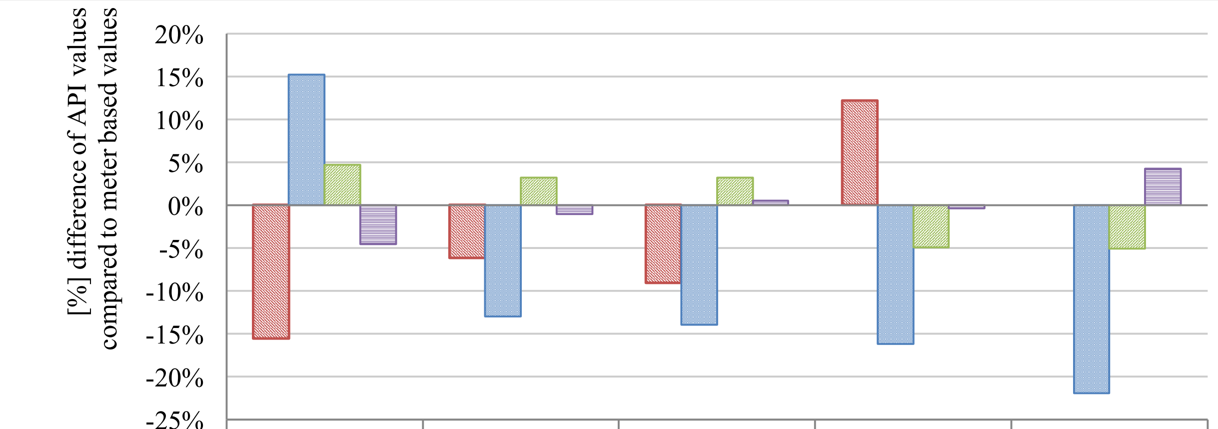
	Power consumption measured using software API		Power consumption measured using volt meter and ammeter	
Sensor	*Average P*	ΔP	*Average P*	ΔP
Accelerometer, Proximity	135.83mW	--	129.76mW	--
Accelerometer, Proximity, Gyroscope	155.85mW	20.02mW	150.90mW	21.14mW
Accelerometer, Proximity, Compass	208.46mW	72.63mW	201.80mW	72.04mW
Accelerometer, Proximity, Gyroscope, Compass	217.06mW	81.23mW	227.82mW	98.06mW
Accelerometer, Proximity, Gyroscope, Compass, 3G	237.67mW	101.84mW	249.68mW	118.92mW

Table 9: Measured consumptions of Samsung Galaxy III GT I9300 sensors

	Power consumption measured using software API		Power consumption measured using volt meter and ammeter	
Sensor	*Average P*	ΔP	*Average P*	ΔP
Accelerometer, Proximity	219.79mW	--	229.79mW	--
Accelerometer, Proximity, Gyroscope	270.69mW	50.9mW	273.58mW	44.29mW
Accelerometer, Proximity, Compass	275.98mW	56.19mW	274.54mW	45.25mW
Accelerometer, Proximity, Gyroscope, Compass	301.46mW	81.67mW	302.71mW	73.42mW
Accelerometer, Proximity, Gyroscope, Compass, 3G	335.46mW	115.67mW	321.23mW	91.94mW

Table 10: Battery runtime

	iPhone 4	Samsung Galaxy II GT I9100	Samsung Galaxy II GT I9100G	Samsung Galaxy III GT I9300
Sensor	***T***	***T***	***T***	***T***
Accelerometer, Proximity	51.06h	11.86h	47.95h	33.81h
Accelerometer, Proximity, Gyroscope	34.31h	6.81h	40.46h	28.4h
Accelerometer, Proximity, Compass	32.17h	6.25h	30.25h	28.3h
Accelerometer, Proximity, Gyroscope, Compass	25.72h	4.71h	26.8h	25.67h
Accelerometer, Proximity, Gyroscope, Compass, 3G	23.57h	4.471h	**24.25h**	**24.19h**



	Accelerometer, Proximety	Accelerometer, Proximety, Gyroscope	Accelerometer, Proximety, Compass	Accelerometer, Proximety, Gyroscope, Compass	Accelerometer, Proximety, Gyroscope, Compass 3G
iPhone 4	-15,59%	-6,12%	-9,03%	12,17%	
SGII GT I9100	15,22%	-12,96%	-13,92%	-16,20%	-21,91%
SGII GT I9100G	4,68%	3,18%	3,19%	-4,96%	-5,05%
SGIII GT I9300	-4,55%	-1,07%	0,52%	-0,41%	4,24%

Figure 22: [%] difference of API values compared to meter based values

8.3 Discussion of the results

The discussion of the results in this section focuses on three main aspects. First, can an average Smartphone be used for a daily sensing, concerning the battery runtime? Second: How precise is the API based measurement, compared to the volt meter an ammeter based measurement? And third: How much does the selected Smartphone model influence the sensor energy usage?

8.3.1 Duration of battery runtime for a daily sensing

We derived the battery runtime for each Smartphone model and for each sensor usage from our measurements, as shown in Table 10. All investigated Smartphones reach even with all sensors and 3G activated a runtime of about 24h, with one exception (Samsung Galaxy II GT I9100, the exception is discussed in subsection 8.3.3). With this result we can conclude an average Smartphone can be used for a daily sensing and context survey.

The chosen scenarios of Smartphone usage are not limiting the general aspect of the measurements. During our measurement the screen of the Smartphone was switched off. Also no phone call was made or no massaging or web surfing was done. These activities are probably done in an everyday Smartphone usage and decrease the battery runtime additionally. Nevertheless, a Smartphone user usually charges the battery overnight. Thus the battery runtime of about 24h still leaves the phone with some energy for those daily activities.

8.3.2 Comparison of measurement method

As mentioned before we choose a dual approach to measure the power consumption of the Smartphone sensor. The first approach was to simply log the Smartphone internal API battery runtime values. The other chosen approach was to connect a volt meter and an ammeter, so the API results can be verified.

After our measurement we not only have the ability to get the power consumption values of different Smartphone sensors but also to examine how accurate the Smartphone energy API is working. Therefore, we compared the meter based values with the API based values. As we consider the meter based values to be more accurate, we used them as base and related the API values to them. The result of the relation can be seen in Figure 22. For the iPhone the API values diverge up to 16% from the multimeter values. The API values of the Samsung Galaxy II (GT I9100) diverge up to 22% and the values from the other Samsung Smartphones (GT I9100G, GT I9300) diverge less than 6%.

8.3.3 Influence of Smartphone model

Three of the four investigated Smartphones have a similar battery runtime under sensor usage while one phone differs greatly in battery runtime from the other Smartphones. With a certain sensor combination, the Samsung Galaxy II (GT I9100) consummates more than 1.4W while the iPhone 4 with the same sensors activated only consummates about 212mW, which is less than a sixth of the power.

We cannot tell which component of the Samsung Galaxy II (GT I9100) consummates the measured 1.4W but most probably it is not the sensor itself. Most of the power has to be distributed as heat from the Smartphone components. The usual sensor ICs are not capable of distributing 1.4W in heat with a passive cooling, but the main CPU of the Smartphone is. Also the area where the processor is located was getting fairly warm. The increased power consumption also correlates with our observation of the reduced battery runtime, shown in Table 10.

8.4 Conclusion on energy consumption

In this chapter we investigated the energy consumption of the Smartphone sensors accelerometer, gyroscope, proximity and compass in four different Smartphone models. We did this to conclude whether an average Smartphones can be used for a daily sensing and context survey. To get the answer to this question we measured the consumption of the sensors and combination of sensors with a volt meter and an ammeter and also with the help of the Smartphone's internal energy API. We also made some measurement with activated 3G connectivity, and we also measured the runtime of the battery when all sensors are activated.

The runtime of three of the four investigated Smartphones with all sensors activated was about 24 hours. Therefore, we can conclude that a modern Smartphone can be used for a daily sensing and context survey. Another finding is the usability of the energy API of Smartphones for energy measurements. The API values differ from the meter values about 22% which enables it for an approximation, but not for a detailed investigation.

9. Conclusion

In this work the use of multiple context sources for context prediction was proposed. Therefore, we investigated four different aspects in the use of multiple context sources, each summarized in the next paragraphs.

First we investigated whether a combination of multiple context sources can really increase the prediction accuracy. We decided to perform an experiment and therefore we also had to propose a certain method of how to combine several sources for prediction. The results of the experiment based on both, real world data and simulated data showed an increase of 4% to 7% of prediction accuracy. In the same investigation it was also shown that the prediction accuracy will not increase or even decrease when multiple sources are combined in the wrong way.

Second we examined on the selection of beneficial context sources. As we found out some combinations of context source will increase the prediction accuracy while others will not, because they irritate each other. We presented a method of how to choose beneficial combinations of context sources and we also did an experiment to verify the presented method.

Third we investigated the effect of stability against sensor disturbances when multiple context sources are used. Robust prediction systems are needed in real life application because disturbances will always occur in sensors. We made an experiment on real world data and simulated data, where we first artificially introduced different amounts of disturbances and afterwards measured the influence on the prediction accuracy. The prediction approach utilizing only one context source lost 35% – 73% of prediction accuracy while the approach utilizing two context sources lost only 0% - 11% of prediction accuracy, when 50% of the history data for the tests were disturbed.

Forth we explored whether a modern Smartphone can be used for a daily context survey when multiple sensors are used. The energy of a Smartphone is limited and thus the use of multiple sensors will reduce its runtime. We did an experiment where we measured the energy consumption of several sensors and combinations of sensors in four different Smartphones. As it turned out most Smartphones have a battery runtime of about 24 hours when all sensors are activated. Only one Smartphone lasted only about 4 hours. This makes the Smartphone a suitable device to collect sensor data for multiple context sources.

Our daily environment gets equipped more and more with sensors, in the form of smart home sensors, sensors in Smartphones or sensors in wearables. This work emphasizes the use of multiple of the available sensors in multiple context sources in parallel, to enable an acceptable application of context prediction.

10. References

[1] Apple Press Info, Apple Previews iOS 9, https://www.apple.com/pr/library/2015/06/08Apple-Previews-iOS-9.html, last checked december 2015

[2] Apple Watch dokumentation, https://support.apple.com/de-de/HT204666, last checked december 2015

[3] Martin, M.; Nurmi, P., "A Generic Large Scale Simulator for Ubiquitous Computing," in Mobile and Ubiquitous Systems - Workshops, 2006. 3rd Annual International Conference on , vol., no., pp.1-3, 17-21 July 2006

[4] Lau, S. L.; David, K., "Movement recognition using the accelerometer in smartphones," in Future Network and Mobile Summit, 2010 , vol., no., pp.1-9, 16-18 June 2010

[5] Rice, A.; Hay, S., "Decomposing power measurements for mobile devices," in Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on , vol., no., pp.70-78, March 29 2010-April 2 2010

[6] Kanhere, S.S., "Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces," in Mobile Data Management (MDM), 2011 12th IEEE International Conference on , vol.2, no., pp.3-6, 6-9 June 2011

[7] Sigg, S., "Development of a novel context prediction algorithm and analysis of context prediction schemes," PhD thesis, University of Kassel, Germany, 2007, http://www.uni-kassel.de/upress/online/frei/978-3-89958-392-2.volltext.frei.pdf, checked December 2015

[8] Altschul, S. F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J., "Basic local alignment search tool," Journal of Molecular Biology Volume 215, Issue 3, October 1990

[9] Kusber, R.; David, K.; Klein, B.N., "A novel future Internet smart grid application for energy management in offices," in Future Network and Mobile Summit (FutureNetworkSummit), 2013 , vol., no., pp.1-10, 3-5 July 2013

[10] Chen, G.; Kotz, D., "A survey of context-aware mobile computing research", Tech. Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000

[11] Woerndl, W.; Huebner, J.; Bader, R.; Gallego-Vico, D., "A model for proactivity in mobile, context-aware recommender systems", Proc. RecSys '11 Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 273-276.

[12] Sigg, S.; Haseloff, S.; David, K., "An Alignment Approach for Context Prediction Tasks in UbiComp Environments," in Pervasive Computing, IEEE , vol.9, no.4, pp.90-97, October-December 2010

[13] Mayrhofer, R. M., "An Architecture for Context Prediction," PhD thesis, Johannes Kepler University of Linz, Linz, Austria, 2004

[14] Mayrhofer, R. M., "Context prediction based on context histories: Expected bents, issues and current state-of-the-art," Proceedings of the 1st international Workshop on exploiting context histories in smart environments (ECHISE'05) at the 3rd Int. Conference on Pervasive Computing, Munich, Germany, 2005

[15] Mozer, M. C.; Petzold, J.; Pietzowski, A.; Bagci, F.; Trumler, W.; Ungerer, T., "Prediction of indoor movements using bayesian networks," LoCA 2005 Oberpfaffenhofen, Germany 2005

[16] Voigtmann, C.; Lau, S. L.; David, K., "A Collaborative Context Prediction Technique," in Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd , vol., no., pp.1-5, 15-18 May 2011

[17] Eagle, N.; Pentland, A.S., "Eigenbehaviors: identifying structure in routine," Behavioral Ecology and Sociobiology, Volume 63, Issue 7, pp. 1057-1066, May 2009

[18] Petzold, J.; Bagci, F.; Trumler, W.,Ungerer, T., "The State Predictor Method for Context Prediction," Adjunct Proceedings Fifth International Conference on Ubiquitous Computing 2003 , Seattle, USA, October 2003

[19] Vintan, L.; Gellert, A.; Petzold, J.; Ungerer, T., "Person Movement Prediction Using Neural Networks," Technical Report, Institute of Computer Science, University of Augsburg, April 2004

[20] Symeonidis, P.; Nanopoulos, A.; Manolopoulos, Y., "Tag recommendations based on tensor dimensionality reduction," in Proceedings of the 2008 ACM conference on Recommender systems, pp. 43–50, 2008

[21] Weiser, M. "The Computer for the 21st Century," - Scientific American Special Issue on Communications, Computers, and Networks, September, 1991

[22] Feder, M.; Merhav, N.; Gutman, M., "Universal Prediction of Individual Sequenzes," IEEE Trans. Inform. Theory 38 pp.1258-1270, 1992

[23] Song C.; Qu Z.; Blumm N.; Barabási A.L., "Limits of Predictability in Human Mobility," Science Magazine, 2010 Feb

[24] McInerney, J.; Stein, S.; Rogers, A.; Jennings, N. "Exploring periods of low predictability in daily life mobility," Mobile Data Challenge by Nokia, Nokia, Newcastle upon Tyne, UK, 2012

[25] Baumann, P.; Santini, S., "On the use of instantaneous entropy to measure the momentary predictability of human mobility," in Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on , vol., no., pp.535-539, 16-19 June 2013

[26] Smith, G.; Wieser, R.; Goulding, J.; Barrack, D., "A refined limit on the predictability of human mobility," in Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on , vol., no., pp.88-94, 24-28 March 2014

[27] Song C.; Qu Z.; Blumm N.; Barabási A.L., "Supporting Online Material for Limits of Predictability in Human Mobility," Science Magazine, 2010 Feb, http://www.sciencemag.org/cgi/content/full/327/5968/1018/DC1, last checked december 2015

[28] Carroll, A.; Heiser, G., "An Analysis of Power Consumption in a Smartphone," in Proceedings of the 2010 USENIX Annual Technical Conference, 2012

[29] Flipsen, B.; Geraedts, J.; Reinders, A.; Bakker, C.; Dafnomilis, I.; Gudadhe, A., "Environmental sizing of smartphone batteries," in Electronics Goes Green 2012+ (EGG), 2012 , vol., no., pp.1-9, 9-12 Sept. 2012

[30] Perrucci, G.P.; Fitzek, F.H.P.; Widmer, J., "Survey on Energy Consumption Entities on the Smartphone Platform," in Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd , vol., no., pp.1-6, 15-18 May 2011

[31] Perez-Torres, R.; Torres-Huitzil, C., "A power-aware middleware for location & context aware mobile apps with cloud computing interaction," in Information and Communication Technologies (WICT), 2012 World Congress on , vol., no., pp.691-696, Oct. 30 2012-Nov. 2 2012

[32] Abowd, G.D.; Dey, A.K.; Brown, P.J.; Davies, N.; Smith, M.; Steggles, P., “Towards a Better Understanding of Context and Context-Awareness,” in Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing (HUC '99), Hans-Werner Gellersen (Ed.). Springer-Verlag, London, UK, UK, 304-307, 1999

[33] Schilit, B.; Adams, N.; Want, R., "Context-Aware Computing Applications," in Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on , vol., no., pp.85-90, 8-9 Dec. 1994

[34] Lieberman, H.; Selker, T., “Out of context: Computer systems that adapt to, and learn from, context,” in IBM Systems Journal, Volume 39, pp. 617-631, 2000

[35] Fitzpatrick, A.; Biegel, G.; Cahill, S.C.V., “Towards a sentient object model,” in Workshop on Engineering Context-Aware Object Oriented Systems and Environments ECOOS, 2002

[36] König, I.; Voigtmann, C.; Klein, B.N.; David, K., "Enhancing alignment based context prediction by using multiple context sources: experiment and analysis," 7th International and Interdisciplinary Conference Context 2011, Springer LNAI6967 Proceedings, Karlsruhe, pp. 159 – 172, Sept. 2011,

[37] König, I.; Klein, B.N.; David, K., "On the Stability of Context Prediction," AwareCast workshop, Ubicomp 2013, Zürich, Switzerland, 8 - 9 Sept., 2013

[38] König, I.; Memon, A.Q.; David, K., "Energy consumption of the sensors of Smartphones," IEEE ISWCS 2013, Ilmenau, Germany, 27 – 30 August, 2013

[39] Haibo Ye; Tao Gu; Xianping Tao; Jian Lu, "B-Loc: Scalable Floor Localization Using Barometer on Smartphone," in Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on , vol., no., pp.127-135, 28-30 Oct. 2014

[40] Das, S.K.; Cook, D.J.; Battacharya, A.; Heierman, E.O., III; Tze-Yun Lin, "The role of prediction algorithms in the MavHome smart home architecture," in Wireless Communications, IEEE , vol.9, no.6, pp.77-84, Dec. 2002

[41] Cook, D.J.; Youngblood, M.; Heierman, E.O., III; Gopalratnam, K.; Rao, S.; Litvin, A.; Khawaja, F., "MavHome: an agent-based smart home," in Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on , vol., no., pp.521-524, 26-26 March 2003

[42] Mozer, M. C., “The neural network house: An environment hat adapts to its inhabitants,” in Proc. AAAI Spring Symp. Intelligent Environments, pp. 110–114, 1998

[43] Van Laerhoven, K.; Borazio, M.; Kilian, D.; Schiele, B., "Sustained logging and discrimination of sleep postures with low-level, wrist-worn sensors," in Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium, vol., no., pp.69-76, Sept. 28 2008-Oct. 1 2008

[44] Lau, S. L.; König, I.; David, K.; Parandian, B.; Carius-Düssel, C.; Schultz, M., "Supporting patient monitoring using activity recognition with a smartphone," in Wireless Communication Systems (ISWCS), 2010 7th International Symposium, vol., no., pp.810-814, 19-22 Sept. 2010

[45] Petzold, J.; Bagci, F.; Trumler, W.; Ungerer, T., “Next location prediction within a smart office building,” in Proceedings of 1st International Workshop on Exploiting Context Histories in Smart Environments at the 3rd International Conference on Pervasive Computing, 2005

[46] Trumler, W.; Bagci, F.; Petzold, J.; Ungerer, T., “Smart doorplate,” Personal Ubiquitous Comput., vol. 7, pp. 221–226, July 2003

[47] Schmidt, A.; Laerhoven, K.V.; Strohbach, M.; Friday, A.; Gellersen, H.W., “Context acquisition based on load sensing,” in Proceedings of Ubicomp 2002, Lecture Notes in Computer Science. Volume 2498., Springer Verlag, 2002

[48] Gellersen, H.W.; Schmidt, A,; Beigl, M., “Multi-sensor context-awareness in mobile devices and smart artifacts,” article in Mobile Networks and Applications October 2002, Volume 7, Issue 5, pp 341-351, 2002

[49] Spice project website, http://www.ist-spice.org/, last checked december 2015

[50] Matrix projec website, http://tmcc.charite.de/forschung/projekte/matrix/, last checked december 2015

[51] EnKonSens project website, http://www.comtec.eecs.uni-kassel.de/enkonsens/, last checked december 2015

[52] König, I.; Beau, P.; David, K., "A new context: Screen to face distance,", 8th International Symposium on Medical Information and Communication Technology 2014, Florence, Italy, 2. - 4. April, 2014

[53] Huichao Liu; Ying Feng; Liguo Zhang, "Transportation mode identification based on smartphone," in Intelligent Control and Automation (WCICA), 2014 11th World Congress on , vol., no., pp.5349-5354, June 29 2014-July 4 2014

[54] AbdelAziz, A.M.; Youssef, M., "The Diversity and Scale Matter: Ubiquitous Transportation Mode Detection Using Single Cell Tower Information," in Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st , vol., no., pp.1-5, 11-14 May 2015

[55] Bloch, A.; Erdin, R.; Meyer, S.; Keller, T.; De Spindler, A., "Battery-Efficient Transportation Mode Detection on Mobile Devices," in Mobile Data Management (MDM), 2015 16th IEEE International Conference on , vol.1, no., pp.185-190, 15-18 June 2015

[56] Petzold, J., “Zustandsprädiktoren zur Kontextvorhersage in ubiquitären Systemen,” PhD thesis, Universität Augsburg, Fakultät für Angewandte Informatik, Germany, 2005, http://opus.bibliothek.uni-augsburg.de/opus4/files/103/dissertation_petzold.pdf, last checked december 2015