MemorySense: Reconstructing and Ranking User Memories on Mobile Devices

Karl Aberer*, Michele Catasta*, Horia Radu*, Jean-Eudes Ranvier*, Matteo Vasirani* and Zhixian Yan†
*EPFL, Lausanne, Switzerland — {firstname.lastname}@epfl.ch
†Samsung Research America, Silicon Valley, USA — zhixian.yan@samsung.com

***Abstract*—The richness of user-centric information gathered by modern devices can be used to keep track of memorable events, therefore acting as a prosthesis of the prone-to-forget human memory. We propose to combine virtual and physical sensors from mobile devices to infer digital memories of user activities in a semi-supervised fashion. In MemorySense, sensor data is processed by a space and energy efficient algorithm to recognize basic activities. We then use semantic reasoning to aggregate these activities into the digital equivalent of a human episodic memory.**

I. Introduction

The amount of digital information produced by users is massive. Thanks to an ever-increasing range of hardware devices whose capabilities soar while their price plummets (e.g., smartphones, smartwatches), the digital trails of the user behaviour, lifestyle, and social relationships are scattered across a collection of unstructured repositories. As such, efficient long-term archival and querying strategies are very difficult to devise. How to scavenge this data, organise it, extract meaningful information, and present it to the user is of paramount importance to tap into the chaotic piles of digital information that the users (semi-consciously) dump online.

Building a lifetime digital store of everything (e.g., articles, books, letters, memos, photos, etc.) on desktop computers is a long-standing challenge that dates back the early 2000. Among other, the MyLifeBits project [1] aimed at fulfilling the Memex vision by developing a system that stores all the digital media of a user, offering visualisations and annotations capabilities. However, the development of a digital memory system on a mobile platform poses unique challenges, and at the same time opens new opportunities. Thanks to the increasing number of embedded sensors such as GPS, accelerometer and microphone, user-centric sensing applications built on smartphones and wearable devices are enabling new ways to sense and understand the way we live (e.g., quantified self) and the world that surrounds us [2]. There are several applications that focus on structuring the digital information and recognising/quantifying specific aspects such as the physical activities or the visited places. See for example [3], [4], [5] and [6]. However, to the best of our knowledge, how to build a system capable of identifying the memorable events of a user, given the signals and heterogeneous data sources that can be obtained from the smartphone, is still an open question. A first step in this direction has been taken by Guo et al. [7]. By using mobile tagging, such as RFID or barcodes, human memories are associated with physical objects, effectively externalising part of the user memories, while at the same time building an object-based social network for memory sharing.

Departing from classical infrastructures that provide static views over heterogeneous sources, we propose to mimic the way our brain stores and accesses information [8] in order to narrow the cognitive burden of accessing digital memories. We focus on collecting and analysing mobile data from smartphones, using multimodal data fusion, semi-supervised online classification, and personalised statistical models to automatically extract user digital memories. In MemorySense, we aim at leveraging the sensors embedded in modern smartphones (i.e., GPS, accelerometer) and combining with external open data sources (i.e., Google Places API, Foursquare API) to answer the questions that are at the root of the creation of memories (*what* occurred during an episode, *where* the episode took place, and *when* the episode happened[8]). The development of MemorySense on a mobile platform poses unique challenges: coping with the limited computational power, storage and battery; filtering noise from the input data; reconstructing memories from partial chunks of information by exploiting temporal, spatial and semantic correlations; ranking the memories according to the user preferences, and representing them in an appealing form. Although part of the resource limitations could be overcome by migrating the computations to a cloud infrastructure, MemorySense has been designed to explicitly avoid any exchange of sensitive user data with third-parties.

The contributions of this paper are the following: (i) a model to map digital data from smartphones to cognitive memories; (ii) a combination of semi-supervised algorithms with rule-based systems to create the basic elements that form these memories; (iii) a framework to reconstruct the memories by temporally, spatially and semantically correlating the basic elements; (iv) the principle of a user-centric ranking function; (v) a conceptual design of a GUI for memory representation, retrieval and navigation.

II. Episodic Memory Model

Defined by Turving in 1942, the episodic memory represents the human system responsible of storing and recollecting past experiences (e.g., had dinner with Alice last Saturday). In [8], Conway theorizes the way in which episodic memories are stored and retrieved while preserving the temporal dimension. MemorySense is based on such model, and maps memories to its hierarchical representation: going from Episodic Elements (EE) to Episodic Memories (EM). This approach has 2 main benefits: (1) the cognitive burden while interacting with the system is reduced, hence facilitating higher memory recall (i.e., MemorySense acts like a "memory prosthesis"), (2) EEs and EMs encourage composability of memories (i.e., we interpret

EEs as atomic elements, and EMs as high-level overviews of the events happened during a longer timespan).

In MemorySense, EEs are characterized by 3 different dimensions that answer the following questions: *What? When? Where?* The location type and its coordinates (i.e., *Where?*) is specified only for geo-localizable events, while all the other events (including those that MemorySense is able to extract automatically from the phone activities) answer at least *What? When?* Therefore, EEs are detailed descriptions of single, atomic events (similar to how Life Events are encoded in the Facebook timeline). On the other hand, MemorySense is able to recognize EMs out of sequences of specific EEs, even when the sequence is incomplete or out-of-order. It is enough for the user to annotate an EE based on her past activity, and then MemorySense will learn to recognize similar sequences in the future. This feature is of key importance to make sure that MemorySense can display memories at different zoom levels (e.g., 1 day, 1 week, 1 month, etc.). Similarly, MemorySense requires a ranking function to be applied on memories, in order to "forget" less relevant events when the application is displaying an overview of a long time range. We are currently running user studies to find out a suitable ranking function, but we already reduced our search space to functions that take into account: (1) location type, (2) frequency of the same event, and (3) user-defined importance (gathered at input time, with a standard Likert scale).

III. Architecture

The MemorySense approach is summarized by the conceptual architecture depicted in Figure 1. Sensor data comes from two different types of sensors: physical sensors (e.g., GPS, accelerometer) and virtual sensors (e.g., phone, sms, camera, apps).

A. Generation of episodic elements

To generate episodic elements, we are interested in the stay points of the user, which are the locations where the user stayed at least for a minimum amount of time to perform an activity. The Android activity recognition API, which provides an easy way to detect if the user is traveling (i.e., in a vehicle, on a bicycle, etc.) is used as a trigger to de-activate the energy-consuming GPS sensor, therefore saving energy while applying a first filtering layer on the data.

The GPS coordinates provided by the location sensing are given as real values and therefore need to be clustered for three reasons: (i) to distinguish whether the user stays within an area considered as a unique location; (ii) to alleviate the inaccuracy of the GPS sensor; (iii) to limit the computational cost of determining the location type by avoiding the categorization of GPS coordinates belonging to a classified location.

We use the clustering algorithm ESOINN, described in [9], which provides the following advantages: (i) it processes input signals in an *online* and *incremental* way; (ii) it is *unsupervised*, as it allows clustering signals without any ground truth; (iII) it does not require the number of clusters as a parameter, but it allows dynamic detection of an *not predefined number of clusters*.

In order for the algorithm to fit the needs of the project, we implemented several modifications. The forgetting mechanism of ESOINN was altered to prevent data loss, which would require manual input from the user, while keeping performance boost. In addition, a density modifier has been included to reduce the complexity of the generated network, making ESOIN suitable to compute on a mobile device. Finally, the classification function was modified to keep consistent cluster labeling throughout consecutive iterations.

The clusters provided by ESOINN are given a type attribute characterizing the function of the location they represent (i.e., park, restaurant, university, etc.). This type is obtained from the Google place API, which provides a list of places that match some given GPS coordinates.

The MemorySense application explicitly asks the user to annotate the places with the description of the activities that were being performed (participatory sensing). Based on the location types discovered in the previous step, the system provides the user with a list of potential activities, sorted by probability of occurring, from which the user can select the activity she is performing. For example, the clustering algorithm can detect that the user is in Hyde park, then the system provides the user with several activities (running, walking the dog, having a pick nick, etc.) and the user manually selects the one she is performing. After a learning phase, the system can detect that the user is recurrently performing the same action at a specific place and will therefore stop asking the user for her input.

In parallel, virtual sensors are also queried without user acknowledgment, albeit not at regular intervals as in opportunistic sensing. Instead, sensor data is acquired every time the user performs an action (user-triggered sensing). At the time of writing, the sensors that are monitored are: call logs, camera, calendar, SMS, browser history. Triggering a virtual sensor generates automatically an episodic element containing the information relative to the sensor, therefore providing additional content to the creation of episodic memories.

B. Creation of episodic memories

Once correctly identified, the episodic elements need to be aggregated into episodic memories. In figure 1, while *Jogging in Hyde park* is directly transformed into the episodic memory *morning jogging*, the *phone call with Alice* and the *lunch at the Thai restaurant* are aggregated into a single memory *lunch with Alice*. The creation of the episodic memory requires expert knowledge about the composition of the underlying episodic elements. It is possible to infer an episodic memory by using a knowledge driven approach such as presented in [10]. This approach allows for recognition of activities (in our case episodic memories), based on the different sub-activities (episodic elements) it is composed of. The main advantage of this approach is that it is resilient to reordering of the sub-activities. For example having a cinema evening with Alice can be modeled as *having dinner* → *watching the movie* → *walk Alice safely back home*. However if the movie is early enough the user may plan to have dinner after the movie without changing the episodic memory in itself. A second advantage is that it allows for the insertion of extra sub-activities without perturbing the recognition (e.g., having an ice-cream before walking Alice home can be included in the episodic memory of having a cinema evening with Alice).

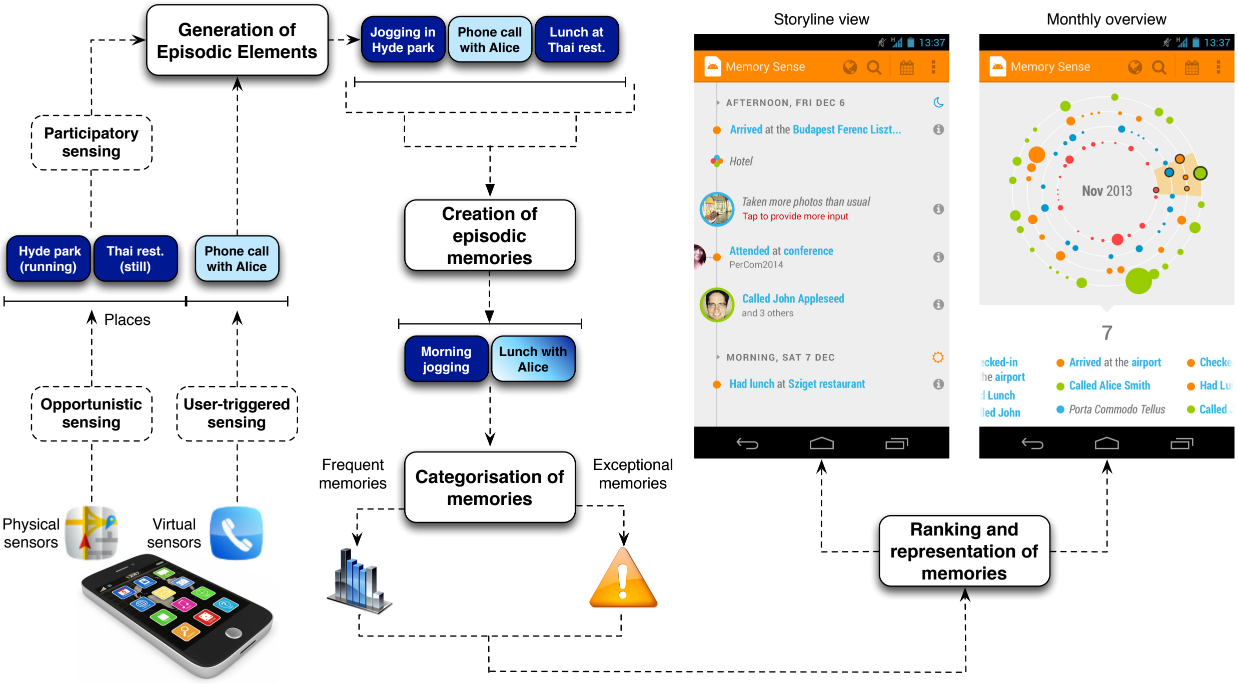

Fig. 1: MemorySense conceptual architecture. The *Storyline view* and the *Monthly overview* are 2 views over the user memories, each focused on a different detail level. The events are encoded as a tuple <*verb, object*>, and categorized with a color (e.g., activity, phone interaction, photos, etc.). Furthermore, in the *Monthly overview*, the size of each color encodes the frequency of a certain event type in the given timespan.

In order to enforce the privacy friendliness of MemorySense, and given that it is not possible to build a model for each activity, the models must be built by the user during the training phase of the system. Once the models are created, they can be used to recognize re-occurrence of activities. The model creation is necessary only for newly discovered activities.

C. Categorisation and ranking of memories

The episodic memories are then categorized as *frequent memories* and *exceptional memories*. Frequent memories are those that occur often in a given time range, such as being at home or training in the gym. Exceptional memories are those that occur with low frequency and may represent something remarkable in someone's life (e.g., a specific holiday in France). The relative frequency notion that is used here is tightly bound to the specific time range. Memories can be very frequent at the week level (e.g., a week of ski sessions in the Alps), while being exceptional if we consider them at the month or year level. For this reason we need to keep track of the distribution of the memories for different levels of granularity. We can then provide a ranking of the memories for a specific granularity, whose score is the average score of its underlying episodic elements. The score of an episodic element is a function of the type of the location, its frequency at the specific granularity, and a 5-star grading coming either from the manual input of the user, in case of new episodic elements), or directly from the system, in case of re-occurring elements.

D. Representation of memories

Finally, memories are showed to the user in a graphical user interface. Several views are envisioned at this stage. The main view consists of a scrolling storyline of the episodic memories of the user, with a zoom function that allows the user to dive into a more specific period. A coarse-grained version of the storyline should display only the most important (in the sense defined in the previous subsection) memories, while the zooming function should provide the user with memories with a lower ranking as well.

A second view focuses on the locations where the user experienced her memories. More specifically, the episodic elements composing this memory are displayed on a map, once again using the zoom function to adjust the number of displayed memories.

The last view that we envision focuses on the meaning of the episodic memories, by grouping semantically-related memories together. This relation can be computed using the activity attached to the episodic memories. From the same view, the user should also be able to define episodic memories by manually grouping episodic elements together. As explained previously, the manual input is only necessary for the first occurrence of an episodic memory. From this input, the system extracts the model of the memories and will use it afterward to detect every new instance of this memory in an automatic way.

IV. Evaluation

We evaluate our approach both in terms of performance and feasibility. The two following experiments are based on the the GPS trace of users extracted from the dataset of the Nokia challenge organized in 2009 [11] [12]. This dataset contains data from 100 users for a period of 10 months. The Nokia

	all	exceptional	frequent
DBSCAN	[0.92 0.89]	[0.71 0.61]	[0.93 0.93]
ESOINN	[0.92 0.86]	[0.95 0.77]	[0.83 0.83]

(a) Average precision and recall [P R]

	1 day	7 days	1 month	6 months	10 months
DBSCAN	0.05s	0.12s	2.55s	14m 12s	36m 8s
ESOINN	2.6s	9s	32s	4m 13s	7m 25s

(b) Processing time for different time windows

TABLE I: Performance evaluation of the location clustering

timeseries contains noisy data resulting from the acquisition of data-points while the users was traveling. While in our architecture, sensor readings are submitted by the Android activity recognition API which detects if the user is traveling, this piece of information is not available in the Nokia dataset. We therefore consider only the datapoints which can be labeled and define the rest as noise which would have been discarded by the Android activity recognition trigger.

A first evaluation aims at assessing the performance of the recognition of episodic elements. We generate a storyline of places, comparing our modified version of the ESOINN algorithm with the state-of-the-art clustering algorithm DBSCAN. The two storylines are compared to a third one based on the ground truth provided by the Nokia dataset. Two alternate storylines are created for each algorithm: (i) one containing only events which are considered exceptional at the scale of a month (i.e., less than 1.5% of events of the current month) and (ii) another one containing frequent events (i.e., at least 10% of events of the current month). The precision and recall values of the storylines of 4 different users have been averaged (see table Ia). Since elements belonging to the true storyline may not appear in the storylines produced by the algorithms, and elements not present in the true storyline may be inserted, the precision and recall computation has been modified by introducing a $null$ class in the confusion matrix. We remark that the precision and recall of the $null$ class is not taken into account in the computation of the overall precision and recall, and only false negatives (i.e., an element is deleted) and false positives (i.e., an element is created) introduced by the $null$ class are factored in the performance metric. While for the storylines containing all events the performance of both algorithms is quite similar, we can observe that our approach outperforms DBSCAN for the exceptional events, and it has a comparable performance for the frequent events.

To evaluate the applicability of our approach in a mobile environment, the two algorithms are ran on an Android smartphone with a CPU of 1.5 GHz quadcore, 2GB of RAM and the operating system Android 4.4. Due to time constraints, we limited our tests to a single user with a GPS trace containing 13674 GPS points in the format <*timestamp, latitude, longitude*>, for which a ground truth is provided. As depicted in table Ib, the processing time for DBSCAN increases dramatically with the length of the trace. Since DBSCAN is not incremental, after 180 days ($\simeq$ 6 months), DBSCAN would need to process all the 179 previous days to generate the daily storyline, with a processing time of 14 minutes and 12 seconds, while ESOINN would need on average 2.60 seconds. Although we envision this daily processing to happen when the smartphone is charging, to limit the impact on the user experience, the growth of DBSCAN processing time will eventually become unpractical and exceed the charging time itself.

V. Current Status and Future Work

MemorySense aims at building a structured representation of the user memories in a semi-supervised fashion, by leveraging physical and virtual sensors in modern smartphones. Such representation exploits Conway's memory model to generate digital memories, similarly to how the human brain generates real memories. This work in progress paper presents the conceptual architecture of the system, including some initial results on the generation of episodic elements and episodic memories. While the initial results are promising, additional work is still required on the aggregation of semantically-related episodic elements, the categorisation of the resulting episodic memories, and their representation.

Acknowledgment

The MemorySense project is sponsored by Samsung Research America, and by Haslerstiftung in the context of the Smart World 11005 (MEM0R1ES) project. Portion of the research in this paper used the MDC Database (owned by Nokia and made available by the Idiap Research Institute, Switzerland). We are thankful to Andrea Peltrin for the beautiful mock-ups.

References

[1] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong, “Mylifebits: fulfilling the memex vision,” in *ACM Multimedia*, 2002.

[2] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, “The jigsaw continuous sensing engine for mobile phone applications,” in *SenSys '10*, 2010.

[3] M. Azizyan, I. Constandache, and R. Roy Choudhury, “Surroundsense: mobile phone localization via ambience fingerprinting,” in *MobiCom '09*, 2009.

[4] N. Bicocchi, G. Castelli, M. Mamei, A. Rosi, and F. Zambonelli, “Supporting location-aware services for mobile users with the whereabouts diary,” in *MOBILWARE '08*, 2008.

[5] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. Legrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. A. Landay, J. Lester, D. Wyatt, and D. Haehnel, “The mobile sensing platform: An embedded activity recognition system,” *PerCom '08*, 2008.

[6] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, “Soundsense: Scalable sound sensing for people-centric applications on mobile phones,” in *MobiSys '09*, 2009.

[7] B. Guo, Z. Yu, X. Zhou, and D. Zhang, “Memphone: From personal memory aid to community memory sharing using mobile tagging,” in *Work in Progress session at PerCom '13*, 2013.

[8] M. A. Conway, “Episodic memories,” *Neuropsychologia*, 2009.

[9] S. Furao, T. Ogura, and O. Hasegawa, “An enhanced self-organizing incremental neural network for online unsupervised learning,” *Neural Networks*, vol. 20, 2007.

[10] L. Chen, C. D. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” *TKDE*, 2012.

[11] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, “Towards rich mobile phone datasets: Lausanne data collection campaign,” *ICPS '10*, 2010.

[12] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, “The mobile data challenge: Big data for mobile computing research,” in *Proceedings of the Workshop on the Nokia Mobile Data Challenge*, 2012.