Supporting Human Memory by Reconstructing Personal Episodic Narratives from Digital Traces

Varvara Kalokyri, Alexander Borgida, Amelie Marian

Department of Computer Science, Rutgers University
v.kalokyri@cs.rutgers.edu, borgida@cs.rutgers.edu, amelie@cs.rutgers.edu

Abstract

Numerous applications capture in digital form aspects of people's lives. The resulting data, which we call Personal Digital Traces - PDTs, can be used to help reconstruct people's episodic memories and connect to their past personal events. This may have several applications, from helping the recall of patients with neurodegenerative diseases to gathering clues from multiple sources to identify recent contacts and places visited – a critical new application for the recent health crisis. This paper takes steps towards integrating, connecting and summarizing the heterogeneous collection of data into episodic narratives using scripts— prototypical plans for everyday activities. Specifically, we propose a matching algorithm that groups PDTs from many different sources into script instances (episodes), and we provide a technique for ranking the likelihood of candidate episodes. We report on the results of a study based on the personal data of real users, which gives evidence that our episode reconstruction 1) integrates well PDTs from different sources into coherent episodes, and 2) augments users' memory of their past actions.

Introduction

Memory plays a fundamental role in life and is critical to our everyday functioning. We use memories to maintain our personal identity, to support our relationships, to learn, and to solve problems. Various new technologies and applications make it possible to digitally capture a huge amount of personal data about every aspect of our lives, such as digital communications we have with friends, events we participate in, trips we make and every piece of digital data we ever create. Our actions result in a multitude of PDTs, kept in various locations in the cloud or on local devices: messages and emails, calendars, location check-ins (e.g. Facebook Places, GPS tracker), online reservations (e.g. Opentable, Ticketmaster), reviews (e.g. Tripadvisor, Yelp), purchase history (e.g. credit card statements), photos, etc. Like a "memex", as envisioned by Vannevar Bush (Bush 1945), this personal data collection can help considerably in remembering everyday events. For example, users could use connections in their data to quickly retrieve digital artifacts, such as minutes from a meeting, pictures from a birthday party, the expense report from a trip, or to help them recall specific memories, such as the names or even faces of people they interacted with, perhaps years ago.

Furthermore, the recent health crisis has highlighted the need for discovering the whereabouts and interactions of people to enable contact tracing. Specifically, contact tracing is an investigative, often laborious process, that involves public health officials interviewing a subject to reconstruct their activities to identify everyone they have been in contact with (Eames and Keeling 2003; Kiss, Green, and Kao 2005; Huerta and Tsimring 2002; Organization et al. 2015)[1]. However, traditional contact tracing is a time-consuming process, depending solely on users' memories. There is evidence (Bradburn, Rips, and Shevell 1987) that humans find it hard to encode and retrieve *routine* experiences. This makes recalling information about where and with whom they have been a challenge. However, this kind of information is being continually recorded (actively or passively) by their digital devices, and could be leveraged to help users recall past actions. For example, a system could help users reconstruct from PDTs the people they have eaten lunch with, the times they went to a grocery store, and the transportation they took.

There has been extensive research in the area of lifelogging, pioneered by (Gemmell, Bell, and Lueder 2006), where the vision is to enable "total recall" of our lives through "total capture" of personally relevant information (Bell and Gemmell 2009; Czerwinski et al. 2006). Such information includes the PDTs we work with: emails, instant texts, web sites visited, bank transactions, etc., as well as other data, such as images, video, and location data. This vision has its detractors (Sellen and Whittaker 2010), who argue that rather than storing a complete lifelog, systems should focus on selecting effective retrieval cues to jog user memories, with the goal of deriving meaning from the collected data.

This work takes steps towards supporting human memory through episodic narratives, an idea based on psychology

[1]Much has been reported recently about "digital contact tracing systems" (Google 2020; Raskar et al. 2020). These are exposure alerting systems that notify users if they were potentially in contact with an infected individual. Exposure alerting and traditional contact tracing go hand in hand in the fight to decrease the spread of COVID-19.

and cognitive science. More specifically, the literature on the psychology of human memory indicates that people have two different kinds of memory: "semantic" and "episodic" memory (Tulving 1972; Conway and Rubin 1993). Semantic memory refers to general knowledge of the world (e.g., you have to pay when buying something) whereas episodic memory refers to the capacity to re-experience specific past episodes (e.g., the occasion you went out to dinner to celebrate your 40th birthday).

We combine representation of both semantic and episodic memory in order to help users recall information of past events. Our approach aims to organize PDTs into episodes, while automatically extracting information about the relevant people and context. This organization will allow creating a personal knowledge base, which users may query to remember a particular event. In addition, our approach aims to provide users with a narrative, which can subsequently be viewed to stimulate memory; this could be particularly useful in a variety of situations, such as for patients with memory difficulties. We believe that our approach will help users recollect aspects of past experiences that have been forgotten, and thereby form a powerful retrospective memory aid.

In this paper, we propose an approach to integrate PDTs from various sources into coherent episodic narratives in order to support human memory. Our approach is centered on the use of so-called *scripts*, first introduced by (Abelson and Schank 1977). Script definitions model dynamic aspects of semantic memory. The paper reviews our conceptual model for describing entities (including PDTs) and scripts, and presents a matching algorithm that groups heterogeneous PDTs into candidate script instances (*episodes*). The PDT types include emails, posts on social media, bank transactions, photos, calendar and location data, etc. Such information can be found in files or extracted through service APIs.[2] The prototypical and non-prescriptive nature of scripts and the sparseness of evidence for each episode lead us to a new, bottom up merging algorithm for episode recognition. This also utilizes a scoring scheme to account for the varied strength of evidence provided by PDTs (e.g., email text vs payment) or script steps. Finally, we report on the results of our approach to episode reconstruction for the `EatingOut` script based on personal data of real users and we show that our approach can successfully integrate PDTs into script instances, and augments users' memory of their past actions, even if these happened less than a month ago.

Personal Data Integration

One of the main challenges in integrating PDTs lies in the fact that data is scattered over many disparate sources, with different data models. To overcome this fragmentation and heterogeneity, one needs a formal conceptual model to represent personal data.

[2]We immediately acknowledge the sensitive nature of this information, and the very important privacy issues that they raise. In our current work, all information obtained resides on an individual's own mobile phone. Users just give Yes/No answers in experiments, without disclosing personal information.

```
{
"message": "Happy birthday John! ☺",                      (WHAT)
"from": "Maria Smith",                                      (WHO)
"place": "Ippudo, New York",                                (WHERE)
"with_tags": "Anna Smith", "John Smith", "George Smith",    (WHO)
"created_time": "2019-05-22T22:43:56+0000",                 (WHEN)
"data_type":"Facebook post"                                 (HOW)
}
```

Figure 1: Simplified Facebook post analyzed according to the *w5h* model

According to the Cognitive Science and Psychology literature, a natural way to remember past events is by any pertinent *contextual* information, which includes answers to the "what, who, where, when, why, how" (*w5h*) questions (Schacter 2002). For example, if you try to remember the name of a restaurant you visited, questions like "When did I go to that restaurant?" and "Who was with me?" will be helpful (Jones 2007).

Our prior work (Vianna et al. 2014) on modeling personal data defined the *w5h* model, which interpreted the six contextual dimensions as follows: What (content), Who (with/from/to whom,...), Where (physical or logical), When (time and date, but also what was happening concurrently), Why (goals, and sequences of events that are assumed to be causally connected), How (application, author, environment).

Figure 1 presents a PDT from a Facebook post, with each piece of information assigned to one of the six proposed dimensions. Information from an e-mail, say, would also be represented using *w5h* (e.g., `from`, `to`, `cc`, would be part of the Who for the email). In both cases, this information can be represented in an ontology-language like OWL using entities and properties both organized in subsumption hierarchies. For example, `from` and `with_tags` would be subproperties of `who` for Facebook posts. The mapping from each source to sub-properties of *w5h* is performed by code requiring an understanding of that service's API.

In addition, since our work focuses on enhancing users' memory of their activities, it is also necessary to have a conceptual model for events, both atomic and complex. For this purpose, we adopted the model we proposed in (Kalokyri et al. 2017b). Recall that our goal for using scripts is to organize PDTs, abstract out relevant information, and help humans remember their events. An example of a script would be "Going out to eat at a restaurant". This script would describe possible "event flows" as shown in Figure 2. The kinds of steps (atomic actions or sub-scripts) comprising a script, and their partial order, are common-sense knowledge we humans learn throughout our lives; e.g., the fact that eating out requires, among others, organizing the outing, possibly making a reservation, getting to the restaurant, paying for the meal, etc. These actions often leave PDTs, which provide *evidence* for that particular event. For example, organizing the outing (e.g., via emails/text messages), possibly making a reservation (e.g., via OpenTable, which confirms by email), getting to the restaurant (e.g., using Uber, which leaves both a payment trace and an email confirmation), paying for the meal (e.g., with a credit card), etc. However, not every step occurs every time one goes out to eat (e.g., some

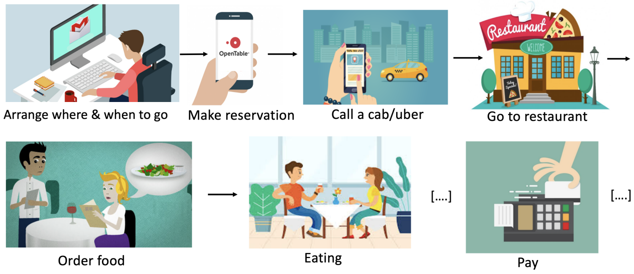

Figure 2: EatingOut script with possible event flows

```
class EatingOut is a SCRIPT{
 locals
  whoAttended < who: set of PERSON
  whereEating < where : EATERY
  whenEating < when : TIME
  whatEaten < what: set of FOODS
  purpose < why : GOAL
 body
  InitiateGoingOut<restaurant>;
  (Loop
     (Or DiscussWhenToEat,
         DiscussWhoWillEat,
         DiscussWhereToEat));
  (Optional
      MakeReservation<restaurant>);
   GoToPlace<restaurant>;
   AttendEatingOut;
   GoToPlace<home>;
}
```

```
class GroceryShopping is a SCRIPT{
 locals
  whoAttended < who: set of PERSON
  whereGroceryIs < where : SUPERMARKET
  whenShopping < when : TIME
  whatShopped < what: set of ITEMS
 body
   InitiateGoingOut<supermarket>;
   (Loop
      (Or DiscussWhenToGo,
          DiscussWhoWillGo,
          DiscussWhereToGo));
   GoToPlace<supermarket>;
   PickItems;
   MakePayment<supermarket>;
   GoToPlace<home>;
 }
```

Figure 3: Definition of EatingOut and GroceryShopping scripts

restaurants do not require or even accept reservations), the order of some steps is not fixed (pay before or after eating), and even more importantly we may find no digital evidence of some steps (e.g., when paying by cash). Since many/most steps of an episode do not leave PDTs, and since scripts only describe prototypical ordering and occurrence of its parts, we do not follow top-down or grammatical approaches to plan recognition, (e.g. as in (Geib and Goldman 2009)), but instead propose a bottom up approach, where we create partially instantiated scripts using individual PDTs as evidence, and then merge compatible instances, accumulating information about it and strengthening evidence for it. This applies to any type of script that the system is told about: “Shopping at the supermarket”, “Visiting a doctor”, “Going on a trip”, etc.

Scripts have properties describing: (i) their goal (for purposes of human explanation); (ii) summary information of the participants in the plan, as well as other descriptive properties, especially *w5h* aspects; (iii) component sub-scripts and atomic actions; (iv) information about (dis)allowed sequencing and timing of sub-scripts/atomic actions. Items (iii) and (iv) describe how the script achieves its goal, and make scripts resemble prototypical workflows.

In the next section, we describe the algorithm for creating episodes (instances of scripts) that a user may have been involved in, based on the PDTs in the user’s database.

Algorithm for Episode Recognition

Our algorithm (see Algorithm 1) starts with the script S we are interested in recognizing, and a large set P of PDTs. Our goal is to create candidate episodes (instances of S) that the user may have been involved in, and relate them to the PDTs.

```
class AttendEatingOut is a SCRIPT{
 locals
   whoAttended < who: set of PERSON
   whereEating < where : EATERY
   whenEating < when : TIME
   whatEaten < what: set of FOODS
   purpose < why : GOAL
 body
   ArriveAt<restaurant>;
   CheckIn<restaurant>;
    (Interleave
       (Or (OrderFood;BeServed),
            SelfServeFood),
       TakePictures,
       PostOnSocialMedia<restaurant>,
       Eat);
   MakePayment<restaurant>;
}
```

```
class MakePayment<T> is a SCRIPT{
 locals
   whoPaid < who: set of PERSON
   wherePaid < where : PLACE
   whenPaid < when : TIME
 body
   (Or PayByCash, PayByCredit,
       PayByCheck);
   (Optional
       ReceiveConfirmationEmail<payment>);

class GoToPlace<T> is a SCRIPT{
 locals ...
 body
   (Optional
      (Or OrderTaxi, OrderUber));
   (Or GoBy<train>, GoBy<bus>, GoBy<car>,
       GoBy<taxi>, GoBy<foot>, GoBy<ship>,
       GoBy<plane>);
  (Or MakeAPayment<train>,
       MakeAPayment<bus>, ...); }
```

Figure 4: Definitions of the AttendEatingOut sub-script and the parametric MakeAPayment<T>, GoToPlace<T> sub-scripts.

Script Syntax: A script specification consists of a top-level (outer) script (e.g. EatingOut script), which we want to instantiate, several sub-scripts and atomic actions, as well as sequencing relationship among them. In addition, every script should contain its *w5h* properties. Figure 3 shows examples of the EatingOut and the GroceryShopping script definitions. Both have local *w5h* properties, and a body of sub-scripts (in colored font) and atomic actions. Figure 4 shows some sub-script definitions: AttendEatingOut, MakeAPayment<T>, and GoToPlace<T>. All (sub)scripts and atomic actions have their own *w5h* properties declared in their definition. For instance, the EatingOut script has *w5h* properties like whoAttended, whereEatingOccured, and whenEatingOccured.

Note that some sub-scripts, such as MakePayment and GoToPlace, are parametric/generic, and are used in multiple places. The argument replacing the formal parameter can be used as in procedures, searching text for a string for example, or the script is written as a case-statement, based on the parameter value; this offers a way to organize and reuse the code for scenario recognition for different scripts.

Now, consider a script that a person has been engaged in, e.g. EatingOut. This script has many events (e.g. invite people, discuss the specifics, make a reservation, go to the place, pay the bill, post photos, etc...). Some of these events provide strong evidence that the person was actually engaged in this script episode. For example, a payment to a restaurant is a strong evidence that the person actually went to it. In contrast, an email mentioning “dinner” or “lunch” is much weaker evidence for planning to go out to eat, and in turn this activity is weaker evidence for having gone out, since the plan may not have been completed, or the event cancelled. Our algorithm uses this idea of strong and weak evidence to rank candidate instances of scripts.

We collect information about strength of evidence the same way we collect commonsense knowledge about the events in a given scenario. We do not see a chance for “objective” ways to score an evidence, as by definition, scripts are imprecise pieces of commonsense knowledge describing some stereotypical human activity, and shared between the members of the same culture.

Algorithm 1: Algorithm for constructing instances of script S

Input: := script definition S; set P of all PDTs
Output: := set $Candidates$ of script instances

D := PDTs with evidence of being a potential instance of script S; **[†]**
$OneStepEpisodes = \{\}; Candidates = \{\}$;
for each d ∈ D **do [††]**
  c_d := new instance of script S, based on d;
  $c_d.score$:= assign score based on strength of evidence;
  $c_d.w5h$:= extract *w5h* information from d and add it to c_d;
  $OneStepEpisodes.add(c_d)$
for all $c \in OneStepEpisodes$ **do [†††]**
  for all $e \in Candidates$ **do**
    if c can merge with e **then**
      e := combine c with e, including $score$ and $w5h$;
  if c not merged **then**
    $Candidates.add(c)$
Use details of script S to look for additional PDTs d that could be relevant to instances in $Candidates$ and repeat **[†††]** ;

```
{
  "EatingOut":{
      "strongEvidence":{
          "MakePayment<restaurant>":0.8,
          "MakeReservation<restaurant>":0.6,
          "PostOnSocialMedia<restaurant>":0.8},
      "weakEvidence":{
          "InitiateGoingOut<restaurant>":0.3}
   },
  "GroceryShopping":{
      "strongEvidence":{
          "MakePayment<supermarket>":0.8}
   }
}
```

Figure 5: Declarative evidence for the EatingOut script.

The declarative description of evidence strength is illustrated in Figure 5. Strong evidence usually includes occurrence of the goal event (AttendEatingOut in this case), which may in turn have its own strong evidence (MakePayment<restaurant>). An example of weak evidence event in this case is InitiateGoingOut<restaurant>.

[†] Retrieving document set D indicating script instantiation: After parsing a script, the next step is to find the set D of documents that provide evidence that an instance of it has taken place. We gather PDTs from various sources using the extraction tool proposed in (Kalokyri, Borgida, and Marian 2018), and use them as "(noisy) sensors" indicating the possible occurrence of corresponding (sub)scripts/atomic actions for which there is strong evidence of having occurred.

For retrieving documents that correspond to an occurrence of some evidence, we must then identify the *clues* to search for in the documents. These clues are either verbs to search for (e.g., "eat", in an email, for identifying InitiateGoingOut <restaurant>) or specific attributes, and metadata that a document may have (e.g., the category in a bank statement should be "Restaurant" vs "Supermarket" for finding evidence for the makePayment<restaurant> vs makePayment<supermarket>). In order to make this easily replicable for various scripts, we consider the *w5h* participants of this script/atomic action, or more specifically its FrameNet frames (Fillmore, Johnson, and Petruck 2003). Then, we use standard sources of synonyms and hyponyms like WordNet and ConceptNet5 (Miller 1995; Liu and Singh 2004) in order to find additional words to search for. The words/phrases in the above generated lists are stored in a text file and are used to retrieve all potentially relevant documents (KEYWORDS_FILE).

```
"makePayment": {
      "restaurant": [{
            "PayByCreditCard": {
                  "how": "plaid",
                  "what": [{
                        "key": "isDebit",
                        "value": "false"
                        }, {
                        "key": "category",
                        "value": ["Fast Food", "Restaurants"]
                        }
                  ]
            },
            "PayByDebitCard": [....]
      }],
      "hotel":[....]
 },
"initiateGoingOut": {
      "restaurant": [{
            "SendEmail": {
                  "how": "gmail",
                  "what": [{
                        "key": "subject",
                        "value": KEYWORDS_FILE
                        }, {
                        "key": "body",
                        "value": KEYWORDS_FILE
                        }
                  ]
            },
            "SendMessage":[....]
      }],
      "theater":[....]
 }
```

Figure 6: Declarative Definition of the Clues for the makePayment and initiateGoingOut scripts.

The result is a list of terms to search for. For EatingOut, the list includes terms like "breakfast", "lunch", "dinner", and "restaurant", plus hyponyms. Figure 6 shows two examples for the clues to search for in the documents, for the MakePayment and the InitiateGoingOut sub-scripts for the restaurant script. In the former, the values to search for are based on the metadata of the document, whereas in the latter script, the KEYWORDS_FILE corresponds to the list of terms produced with the process described above.

We point out the ease of extending the system to recognize new scripts. For example, for makePayment<hotel>, one would just change the value "Hotel" for the key "category" of a bank transaction; for initiateGoingOut<theater>, one would specify a KEYWORDS_FILE starting from "theater" and having terms extracted from WordNet, ConceptNet, etc. The rest of the algorithm remains the same, since the queries for retrieving relevant documents are based on this declarative definition of the clues.

Finally, the set of documents D is pre-processed by: (i) explicating information (e.g., terms like "tomorrow", "on Friday", are made absolute dates using the Natty date parser (Stelmach 2016)); (ii) performing entity resolution for people and places (who and where dimension) using Stanford's Entity Resolution Framework (Benjelloun et al. 2009); (iii) grouping certain kinds of documents (e.g., related email threads, or related sequences of tweets) into a single individuals d in D; (iv) finding the places/venues that the user has visited from the geo-location coordinate history (gps) (Li et al. 2008).

[††] Creating initial script instances c_d: Each document d in D instantiates the corresponding atomic action/subscript, which leads to the creation of a candidate instance of the outer script S in a bottom-up fashion. In addition, every *w5h* property is propagated from the document into the atomic actions and then into the script hierarchy above.

[†††] Merging script instances: A distinctive feature of our system is the presence of multiple sources of evidence for the same script instance. In order to merge them and find what they correspond to in the real episode, every script needs to have *"keys"*, a rating of how well *w5h* (sub)properties identify instances. For EatingOut, keys are whereEatingOccurred, whenEatingOccurred, and, to a lesser extent, who. The what and how properties of this script are not important because they would often lead to incorrect merging (e.g., two instances of eating sushi (what) need not be merged). However, for other types of scripts, e.g. GoingToDoctor script, "what" is more important than when and where (i.e. we would want to merge all the times we went to a doctor for a specific issue/disease (what)). Since every script can have different keys, this has to be explicitly mentioned in the algorithm. This information can be acquired by commonsense acquisition knowledge techniques. When two instances share the same/similar "keys" (some keys, such as time and place can be assessed for similarity using distance) they become candidates for merging. For merged pairs, the *w5h* property fillers are unioned, and the score for the merged instance is computed, using Hooper's rule (Shafer 1986) for combining probabilistic evidence: $score(e) = 1-(\prod_{e \in Candidates} 1-score(e))$ where $score(e)$ is the score for the merged instance e.

Note that the above cannot be done in a single step: having established that a script instance is occurring with a certain degree of certainty, additional PDTs can be gathered as part of the instance when examining the script definition. For example, an Uber receipt might be added to the script instantiation as a result of the "GoToPlace<restaurant>" sub-script once an instance of EatingOut has been created. Note that this sub-script could not initiate on its own an episode of EatingOut, since Uber receipts can be part of many different kinds of episodes if we don't know the precise destination. As part of future work we plan to dynamically learn the scoring function based on user's data and relevance feedback.

Complexity Analysis

The complexity of the first part (†), is essentially the complexity of optimized database search for clues to every (sub)script step, whose "data complexity" is $O(\log size(P))$. If $n = size(D)$, then part (††) is $O(n)$. In the worst case, (†††) fails to merge anything at each iteration, so all pairs of original $OneStepEpisodes$ from (††) are compared, resulting in $O(n^2)$ work. Note that n is much smaller than $size(S)$ (e.g., from our previous study (Kalokyri et al. 2017a), $size(P) = 5,282$, whereas $n = 109$).

Script Acquisition

Scripts are a form of commonsense knowledge, which is seen to be at the frontier of AI/NLP (Sap 2020). There has been extended literature in the field, such as acquiring commonsense knowledge from the general public (Singh et al. 2002), the use of games for that purpose (Von Ahn and Dabbish 2008), and the use of crowdsourcing (e.g. (Wanzare et al. 2016) gathers data for 13 scripts and attempts to align each set semiautomatically). Each of us is capable of providing a first pass at partially-ordered steps required to visit a doctor, say, but it is more work to find PDT evidence (strength) for the steps, and much harder to be *comprehensive*. We have experimented with Mechanical Turk crowdsourcing for this, but script acquisition is the focus of separate work (see (Wanzare 2020) and references therein for earlier work in the field) and is beyond the scope of this paper.

Generalizability

In addition to acquiring new scripts, we are also working on developing families of related scripts by *incrementally modifying more general scripts through inheritance*: e.g., from GoingOut4Entertainment to GoingOutRestaurant, GoingOutSportsEvent, GoingOutTheater, etc. (e.g., (Borgida, Kalokyri, and Marian 2019)). A contribution of this paper is a software architecture which supports systematic and declarative construction/extension. Topics mentioned include: (i) scripts being parametric/generic: The parameter can be used as in procedures, searching text for a string for example, or writing the script as a case-statement, based on the parameter value. (ii) suggested standard sources of synonyms and hyponyms (e.g., WordNet, ConceptNet5) for signaling words; (iii) proposed use of FrameNet to determine w5h participants of events and to find additional search words; (iv) declarative representation of evidence to look for in an episode (Fig. 5) and (v) declarative definition of the clues to search in PDTs (Fig. 6).

Experiment Design

To evaluate the efficacy of our approach we ran experiments on real users' data, where our goal was to find instances of them going out to eat at restaurants. We used this script example because it generates a variety of PDTs, and is similar to many other entertainment scripts such as going to a theater, going to a concert etc. It is important to mention that performing experiments on Personal Data is difficult due to the sensitive nature of the data, the difficulty of getting personal datasets as well as having a baseline for comparisons.

For this reason we developed a mobile app [publicly available[3]] which users are able install on their devices and which can also be used for future research purposes. Note that we did not have access to the users' data. The users just reply Yes/No answers without disclosing any personal information. The tool is made for Android 7.0 version or later and the PDTs collected are from the following services:

- Messaging: Messenger, Phone text messages
- Social Media: Facebook, Instagram
- Email: Gmail
- Calendar: Google Calendar
- Financial Data: Plaid API, directly downloaded .csv files from bank institutions
- Location Data: Google Maps location history, GPS data
- Photos: Google Photos

Procedure

We recruited participants using flyers and email lists at Rutgers University. We required the participants to be at least 17 years old, to be active Android users, and to communicate in English. The participants were compensated with cash for completing the whole study. The study was approved by the Rutgers University IRB committee.

Prospective participants interested for our study had to sign up and take a survey in order to assess their familiarity and usage with the services supported by our application, as well as if they were active users (as concerning going out to eat at restaurants). They were asked to reply through a Likert scale (1-5) and full text answer about:

- how much they think they use each of the services that the app supports
- how often they go out to eat at restaurants per month
- what services/apps they use to make plans to go out to eat, make reservations, get reminded of outings, pay and checking in at restaurants (i.e. the sub-scripts of the EatingOut script)
- which services they are willing to give access to the app.

[3]https://github.com/yourdigitalself

We received 42 responses in total. Figure 7 shows the different sources reported by users for the five main sub-scripts of EatingOut. A first observation is that users clearly behave differently. Most use either plain text messages or phone calls to arrange to go out to eat. Other users use Messenger, Instagram messages or emails in order to communicate with their friends concerning this matter, and a smaller percentage uses Snapchat, Whatsapp and Groupme. On the other hand, users seem to agree in the way they make reservations (by phone) and pay at restaurants (with credit card or cash). In addition, users seem to not write down their restaurant outings in a digital form. Finally, they seem to use many different sources for letting their friends know that they are at or went to a restaurant. The majority claims they use Instagram first and then Facebook, whereas 31% claims that they do not use any online service.

These results show that looking at several sources of PDTs to identify script instances for a given user is critical, and that any approach to retrieve user memories must consider multiple sources of PDTs to adapt to the wide variety of user behaviors.

Experimental Setup

Out of the 42 recorded responses, we selected 16 participants for an in-depth study. Participants were selected based on their interest, willingness to use the app, use of services included in the app, and frequency of restaurant outings (at least 5 per month). Out of the 16 participants, 9 were male and 7 female, between ages 19 and 49, and the study was carried out with one month's data. The steps that we followed were the following:

1. Before the experiment, we asked participants to try to remember the occasions of them going out to eat at restaurants and then carefully go over their past month's digital information, and add any missing outings, including name of the restaurant – where, date they went – when, with whom they went – who. We used this information as a proxy for recall.
2. We introduced the participants to the experiment, and we installed the app on their phone.
3. Participants were asked to give permission to the app to download one month's PDTs of services they wanted.
4. Participants were shown all candidate script instances of them going out to eat at restaurants, and had to indicate Yes/No for each of the instances, with further Yes/No questions as follows:
    - Yes: evaluating the *w5h* information deduced
        1. Who: All the identified people are correct, but there are some people missing.
        2. Who: Some of the identified people are incorrect.
        3. Who: All the identified people are incorrect.
        4. When: The date identified is wrong.
        5. Where: The place name is wrong.
        6. Other. Please specify.
    - No: choose a reason why not.
        1. This is not a restaurant.

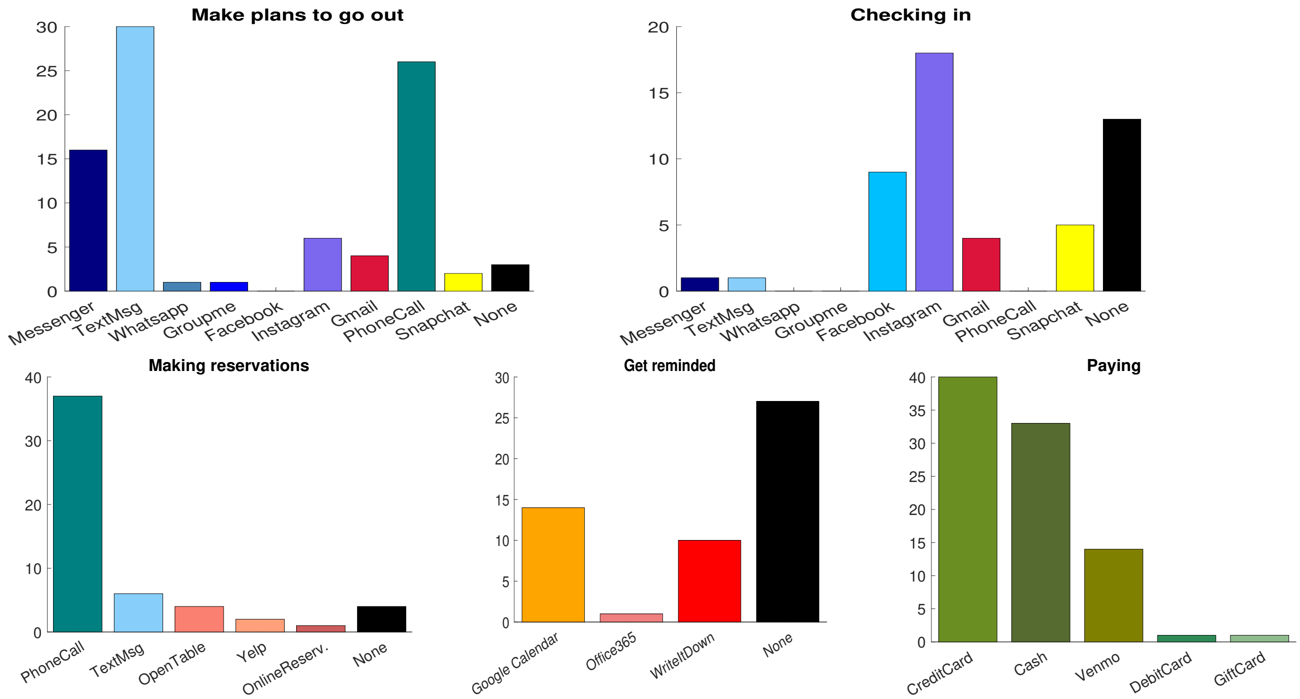

Figure 7: Sources used by users for the EatingOut script. *Top-Left:*Make plans to go out eat, *Top-Right:* Let friends know that they are at a restaurant, *Bottom-Left:* Make reservations at restaurants, *Bottom-Middle:* Get reminded of restaurant outings, *Bottom-Right:* Pay at restaurants

   2. Take out order at that restaurant.
   3. Someone else went to that restaurant.
   4. It is a restaurant but I didn't go there to eat.
   5. Other. Please specify.
5. We uninstalled the app from the users' phones and destroyed the associated data.
6. We asked participants' opinions and comments at the end of the experiment.

Experimental Evaluation

We now detail the results of our evaluation, looking at the quality of the script instance recognition.

We evaluate our matching algorithm and scoring function, by reporting two kinds of results: (1) how well our approach recognized script instances, and (2), how well it recognizes and abstracts When, Where and Who information from subscripts and atomic actions into the (outer) script instance. For both cases, we report two different kinds of relevance:

1. **Binary Relevance**
   - A proposed *script instance* is judged *relevant* if the user did go out to eat at a restaurant, even if the *w5h* information was only partly correct. Cases like takeout were counted as false positives in this experiment.
   - Each w5h *information* is judged *relevant* if it is exactly correct (subset or superset does not count). For example, if in the Who dimension only a subset of people that attended is recognized, then this would be irrelevant.
2. **Graded Relevance**

   A proposed script instance is:
   - *Exactly relevant:* when the user actually went out to eat at that restaurant.
   - *Relevant but too broad:* when a restaurant outing is identified, but the user didn't go there to eat (e.g. went for dancing, just hanging out with a friend etc).
   - *Relevant but too narrow:* when a restaurant outing is identified, but the user didn't stay at the place to eat (e.g. it was a takeout order).
   - *Partially relevant:* when a planned outing was correctly identified, but the user didn't end up going in the end.
   - *Not relevant:* when the identified instance is not about a restaurant.

   We assume that a piece of (when, where, who) *w5h* information is:
   - *Exactly relevant:* when the all identified *w5h* information is correct.
   - *Relevant but too broad:* when the *w5h* information contains relevant information but also includes other irrelevant information.
   - *Relevant but too narrow:* when the *w5h* information contains relevant information but is lacking some information.
   - *Not relevant:* no relevant information.

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
Our approach	13	15	10	5	14	24	13	6	19	17	19	7	11	5	17	15
User memory	7	8	8	5	8	10	5	5	9	9	13	5	6	3	8	9
User data	13	14	11	5	7	14	11	6	20	14	15	5	11	5	16	14
Recall	1	1	0.91	1	1	1	1	1	0.95	1	1	1	1	1	1	1

Table 1: Number of identified EatingOut actions by users vs number of correct events our approach retrieved per user as a proxy for Recall

Metrics

Based on both the binary and graded relevance, we report on the following metrics:

- **Percentage of instances retrieved:** the percentage of all EatingOut events identified by users which were retrieved by our scripts; a proxy for Recall. [4]
- **Mean Average Precision @ k (MAP@k):** MAP as a binary relevance assessment for the percentage of the top-k identified script instances that correspond to actual EatingOut events. We assumed the result counted as true positive only if the users annotate a search result as "Exactly Relevant". The same holds for the *w5h* information.
- **Normalized discounted cumulative gain (nDCG):** nDCG for assessing the ranked results when taking into consideration graded relevance, as described above. For our experiments, we translate the five grades of relevance as follows: Exactly relevant has a score of 5, Relevant but too broad and Relevant but too narrow have a score of 4 and 3 respectively, Partially relevant has a score of 2, and Not relevant has a score of 1.

Experimental Results

Our results allow us to make several observations.

Routine experiences are hard to retrieve. Table 1 shows the number of correct EatingOut instances retrieved by our approach compared with the number identified by users from memory, and by searching their PDTs (proxy for Recall). A first observation is that the results clearly indicate how hard is for users to recall their outings in the previous month, either from memory, or even when asked to go through their digital information. Our tool identified more correct instances than the users were able to recall and find in all cases but two (user 3 and 9), where the recall was 0.91 and 0.95 respectively. We anticipated this for two reasons. First, there is evidence (Bradburn, Rips, and Shevell 1987) that routine experiences, like going out to eat, are less likely to be encoded and harder to be retrieved, whereas unique experiences are particularly likely to be encoded and recalled. Second, users had a hard time reviewing their digital information since they had to look in so many different services. Our tool identified more instances than the users were able to identify by searching their PDTs in most of the cases. Users found it hard to search through their data when using only messages to arrange an outing, paid by cash, or without having their GPS activated, since most of the applications have keyword-based search. Users 5 and 6, who were able to retrieve only half of their outings, clearly show this issue.

This finding supports how helpful our system can be not only for supporting human memory but also for contact tracing for epidemiological purposes. It can help users remember when they went out to eat, where and with whom. Similarly, using the grocery shopping script, the system could reveal if a user took the train/bus to go shopping (if this was instantiated from the "GoingToPlace" sub-script), the times they went, and to which grocery stores.

Quality of information given by different sources vary. Table 2 shows the overall precision of the identified script instances along with the sources each user incorporated in the application. Our approach achieves a total of 78% for all the users. User #2 achieved the highest precision of all, since the sources they chose to include in the study contained bank transactions, google maps location history, instagram and facebook, sources that tend to be of high quality, whereas User #8 achieves the lowest precision of all, since they included their private phone text messages without any high quality source, such as location or bank data. The reason why text documents tend to be of lower quality is because they depend on keyword matching for relevance.

The results above show that our scripts achieve good precision. However, retrieval systems typically return results in a ranked order, and users are expecting the first few results to be the most relevant. We now look at the quality of the returned answers by evaluating the Precision@k metric.

Quality of the returned answers. Figure 8 shows the Mean Average Precision@k for all the identified script instances for all the users. As shown, our approach achieves a really good precision even for low values of k. The reason for that, is that our approach does include many different kinds of sources and is able to account for all the different kinds of user behavior. In addition, the users seem to use different services for carrying out different actions of a particular script instance, as noted in Figure 7.

Figure 9 shows the normalized discounted cumulative gain (nDCG) for the ranked results when taking into consideration the graded relevance. The nDCG was computed by normalizing the DCG@k with the ideal DCG value or IDCG@k. Hence, we computed the IDCG at each level k

[4] Ideally the ground truth could be constructed by asking users to journal their lives over a period of time. However it is unclear whether the mere act of journaling would have an impact on the type of data found in the users' personal data - would the user record more information as a side effect of journaling? - and lead to a case of observation bias.

	Sources	**Precision**
User #1	Social Media, Calendar, Financial Data, Location Data, Google Photos	0.87
User #2	Social Media, Location Data, Financial Data	**0.94**
User #3	Email/Messaging, Social Media, Calendar, Financial Data	0.66
User #4	Email/Messaging, Social Media, Calendar, Financial Data, Location Data, Google Photos	0.66
User #5	Email/Messaging, Social Media, Calendar, Financial Data, Location Data	0.74
User #6	Social Media, Calendar, Financial Data, Location Data, Google Photos	0.89
User #7	Email/Messaging, Social Media, Calendar, Financial Data, Location Data, Google Photos	0.76
User #8	Email/Messaging, Social Media, Calendar	**0.6**
User #9	Email/Messaging, Social Media, Calendar, Financial Data, Google Photos	0.74
User #10	Email/Messaging, Social Media, Calendar, Financial Data, Location Data	0.81
User #11	Email/Messaging, Social Media, Calendar, Financial Data, Location Data, Google Photos	0.76
User #12	Social Media, Calendar, Financial Data, Location Data, Google Photos	0.86
User #13	Email/Messaging, Social Media, Calendar, Financial Data, Google Photos	0.73
User #14	Email/Messaging, Social Media, Calendar, Financial Data, Location Data, Google Photos	0.83
User #15	Email/Messaging, Social Media, Calendar, Financial Data, Google Photos	0.77
User #16	Email/Messaging, Social Media, Calendar, Financial Data, Location Data, Google Photos	0.79
Total		**0.78**

Table 2: Overall precision for each user

	when	where	who
MAP	0.85	0.81	0.21

Table 3: MAP for when, where, who dimensions for all users

and then computed the average nDCG across the 16 results. It is clear that our ranking quality is high, and our approach is able to recognize and distinguish highly relevant PDTs in favor of irrelevant PDTs.

We then report the same metrics (MAP, MAP@k and nDCG@k) on the when, where and who dimensions.

The who dimension is harder to be retrieved. Table 3 shows the MAP for the three dimensions for all users. A first observation is that the when and where dimensions are easier to extract than the who dimension due to the metadata that the PDTs have and due to the fact that if there is a payment or a gps location for an outing these two dimensions are easier to get extracted. On the other hand, the who dimension is more difficult to get extracted for the following reasons. Most of the participants' restaurant outings as shown in Figure 7, were arranged either by phone or by text messages. Our system does not capture voice data, so it was anticipated that we will miss some information about the phone arranged outings[5]. In addition, the participants mentioned that many of their outings were organized on the fly, by talking to each other in person, while in work, or while being with friends. Our evaluation also penalizes correct but incomplete who dimensions by counting them as false positives.

Figure 10 shows the MAP@k for the three dimensions for all the users. As shown, our approach achieves a good precision for the when and where dimension for all the values of k. On the other hand, the precision for the who dimension drops as k increases. This happens due to the fact that for low values of k, the score of the instances is low, which means there are not many PDTs to account for these instances. In that case, as previously mentioned, the who dimension is the hardest dimension to be retrieved, because either there is no information recorded or our approach either lacks or recognizes more people in an outing. This is actually demonstrated in Figure 11, which shows the nDCG for the ranked results when taking into consideration the graded relevance. We can now observe how much better the accuracy is for the who dimension getting a gain of 0.7 as the highest and 0.5 as the lowest value. This means that our approach is able to recognize some people in the outing, but it's hard to recognize them all correctly. In addition, we observe that the when dimension is getting extracted with a better accuracy than the where dimension. This is mainly because the when dimension can be extracted by all the PDTs, whereas the where dimension can be missing from messages, emails or photos.

Our system was rated positively by the participants. In general, users were satisfied with the efficiency of use of the application and the functionality it offers. The majority was excited about using the app. They mentioned how "cool" it is to be able to see their data organized, and how easy it is to navigate to their data, since it is linking to the original piece of information. In addition, some were surprised when they found out some outings that they had totally forgotten. They pointed out that this happens to them frequently, and that although they do remember having some pieces of information somewhere in some service, they don't know where to search, or what to search. Finally, some mentioned that they

[5]As part of future work, we plan to retrieve phone numbers from google maps, (in case it is already recognized the name of the venue, i.e. "where"), and associate them with calls that the user might have made on these numbers, thus providing corroborating evidence. In addition, we plan to incorporate conversations from personal assistants like Google Now and Alexa which record users' commands and replies.

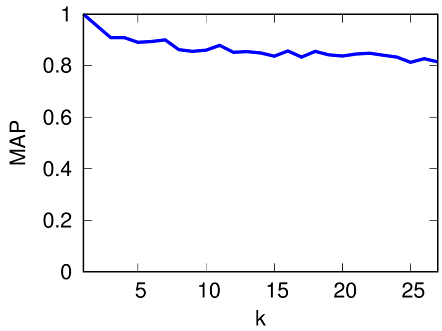

Figure 8: MAP@k for the recognized instances

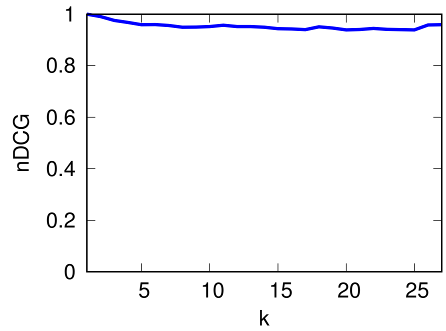

Figure 9: nDCG@k for the recognized instances

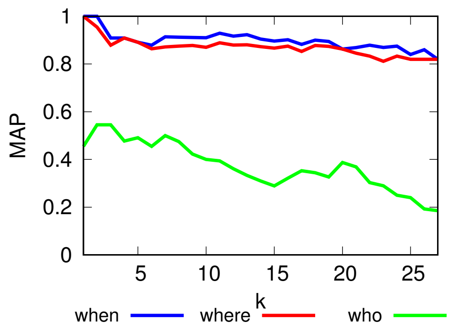

Figure 10: MAP@k for when, where, who dimensions

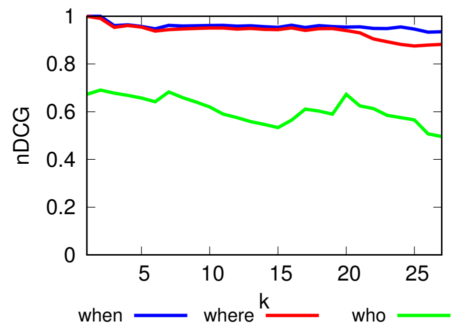

Figure 11: nDCG@k for when, where, who dimensions

would prefer to be able to search through our app, rather than browsing, as well as they wished that some other services were supported by the app (i.e. Snapchat, WhatsApp).

Limitations of the study. The number of participants was small, and they were from an academic environment, thus possibly not representative. Also, we presented results for one script scenario. As future work we plan to conduct more experiments with a variety of scripts, larger population and participants from different backgrounds.

Related Work

We review the many areas of related work, some of which have already been mentioned briefly in the previous sections.

Activities of Daily Living, Pervasive Computing, LifeLogging and Memory Tools. The use of aids to help people with memory deficits is thought to be one of the most effective ways to aid rehabilitation (see (Kapur, Glisky, and Wilson 2004) for a review). Most external memory aids focus on improving prospective memory; they help people to remember to keep appointments, take medication, etc. Available devices include calendars, alarms, Post-it notes, as well as more sophisticated systems, like Amazon Alexa, Siri, and Google Now (Hoy 2018; Thakur 2016). However, these systems' purpose is to remind users of events based on their personal data, often with some commercial goal, and are limited to using information that is in the vendors' proprietary systems. In addition, these systems focus on prospective tasks: remembering to carry out tasks either based on a time or event trigger; while our current application scenarios are centered around retrospective tasks: organizing past memories. In contrast, there are few memory aids designed to improve the ability to remember past experiences. Perhaps the two most obvious examples are cameras and diaries. Sensecam (Hodges et al. 2006), and Kalnikaite's browser (Kalnikaite et al. 2010) which adds GPS, are tools closely related to ours used for the recall of everyday events by passively recording images combined with GPS and relating them to everyday activity in order to trigger autobiographical memory in people with memory issues. Other tools include the MemoClip , the Cyberminder, and Memory Glasses (Beigl 2000; DeVaul, Pentland, and Corey 2003; Dey and Abowd 2000). Our work is distinguished from most of the above efforts by the fact that we use the vast amount of existing digital traces already being produced, rather than capturing new data. Our data integration approach could however be used in conjunction with prospective approaches to personalize activities recognition.

In addition, the area of life-logging is quite similar, and is surveyed in (Gurrin et al. 2014; Van Den Hoven, Sas, and Whittaker 2012). Bell has pioneered this field with MyLifeBits (Gemmell, Bell, and Lueder 2006) for which he digitally captured all aspects of his life. (Naeem, Bigham, and Wang 2007) uses the plan description language Asbru (designed to describe medical protocols as skeletal plans) to recognize activities of daily living from kitchen sensors. A particularly relevant paper is (Meditskos et al. 2018) which used the technique of multi-sensory data analysis along with egocentric video recording from a bracelet to aid the dementia patients recognize their daily living activities. Our approach is distinguished from the above efforts by the facts that: (i) there is massive concurrent execution of different scripts; (ii) many PDTs are not part of any script execution; (iii) the execution of some script steps is not manifested in PDTs; (iv) script instances are often exceptional variants of the prototype.

PIM. The general research area of Personal Information Management (PIM) began in the '80s to help users better store, integrate and query large collections of varied digital data. Researchers have suggested PIM interfaces for web activities (Dumais et al. 2003; Kaptelinin 2003; Murakami and Mitsuhashi 2012), email (Ayodele, Akmayeva, and Shoniregun 2012; Whittaker, Bellotti, and Gwizdka 2006), and local files (Barreau and Nardi 1995; Barreau 1995). The central focus of such systems is the identification of relevant objects in the user's information space, and establishing their interrelationships. Often this is based on a domain or personal ontology. In contrast with this static view of information, we focus on a dynamic approach to the integration of PDTs, by providing a narrative to make connections between them.

Processes and Plans. Since scripts are plans, and we recognize plan instances from PDTs, the extensive literature on plan recognition, such as (Geib and Goldman 2009; Geib, Maraist, and Goldman 2008) is obviously relevant. One important difference is that these approaches start from a description of a domain in terms of planning operators, while scripts are pre-compiled stereotypical plans. Also, closely related is the area of activity recognition, which often considers the problem of recognizing lower-level actions, of which plans are composed, especially when these are signaled by sensors. Our needs include the ability to recognize multiple, concurrent, and interleaved script instances and components. Most importantly, our situation is distinguished by the fact that most of the PDTs we encounter do not signal any script, and a large fraction of steps in any particular instantiation of a script leaves no trace ("missing actions").

Conclusion

We have presented a novel script-based approach to integrate and connect heterogeneous collections of PDTs into coherent episodes of user activities, which extract relevant summary information, as well as a software architecture that supports systematic and declarative specification of evidence. Our approach can help users explore their events in an integrated way by creating a personal knowledge base they can access and search in the future. Experiments on real users' personal data for the script of going out to eat showed that our approach augments people's memory for past actions, which can subsequently help them to stimulate their memory. Simple variants of EatingOut, such as GoingToTheater and other forms of entertainment, would cover many more cases. This can be particularly useful in a variety of situations such as people with memory deficits, as well as contact and location tracing. In addition to its applications to personal data management and memory augmentation, our work provides opportunities for behavioral researchers to study user behavior patterns. For example, the combination of PDTs, AI, and self-reported surveys can yield new insights into mental health assessment.

References

Abelson, R.; and Schank, R. C. 1977. Scripts, plans, goals and understanding. *An inquiry into human knowledge structures New Jersey*, 10.

Ayodele, T.; Akmayeva, G.; and Shoniregun, C. A. 2012. Machine learning approach towards email management. In *World Congress on Internet Security (WorldCIS)*. IEEE.

Barreau, D.; and Nardi, B. A. 1995. Finding and reminding: file organization from the desktop. *ACM SigChi*, 27: 39–43.

Barreau, D. K. 1995. Context as a factor in personal information management systems. *Journal of the American Society for Information Science*, 46(5): 327–339.

Beigl, M. 2000. MemoClip: A location-based remembrance appliance. *Personal Technologies*, 4(4): 230–233.

Bell, G.; and Gemmell, J. 2009. *Total Recall: How the e-Memory Revolution Will Change Everything*. Dutton. ISBN 978-0-525-95134-6.

Benjelloun, O.; Garcia-Molina, H.; Menestrina, D.; Su, Q.; Whang, S. E.; and Widom, J. 2009. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 18(1): 255–276.

Borgida, A.; Kalokyri, V.; and Marian, A. 2019. Description Logics and Specialization for Structured BPMN. In *Int. Conf. on Business Process Management*, 19–31. Springer.

Bradburn, N. M.; Rips, L. J.; and Shevell, S. K. 1987. Answering autobiographical questions: The impact of memory and inference on surveys. *Science*, 236(4798): 157–161.

Bush, V. 1945. As We May Think. *The Atlantic Monthly*.

Conway, M. A.; and Rubin, D. C. 1993. The structure of autobiographical memory. *Theories of memory*, 103: 137.

Czerwinski, M.; Gage, D. W.; Gemmell, J.; Marshall, C. C.; Pérez-Quiñones, M. A.; Skeels, M. M.; and Catarci, T. 2006. Digital memories in an era of ubiquitous computing and abundant storage. *ACM*, 49: 44–50.

DeVaul, R. W.; Pentland, A.; and Corey, V. R. 2003. The memory glasses: subliminal vs. overt memory support with imperfect information. In *Symp. Wearable Computers, 2003*.

Dey, A. K.; and Abowd, G. D. 2000. Cybreminder: A context-aware system for supporting reminders. In *Int. Symposium on Handheld and Ubiquitous Computing*, 172–186.

Dumais, S.; Cutrell, E.; Cadiz, J. J.; Jancke, G.; Sarin, R.; and Robbins, D. C. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. In *(SIGIR'03)*.

Eames, K. T.; and Keeling, M. J. 2003. Contact tracing and disease control. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1533): 2565–2571.

Fillmore, C. J.; Johnson, C. R.; and Petruck, M. R. 2003. Background to framenet. *International journal of lexicography*, 16(3): 235–250.

Geib, C. W.; and Goldman, R. P. 2009. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11): 1101–1132.

Geib, C. W.; Maraist, J.; and Goldman, R. P. 2008. A New Probabilistic Plan Recognition Algorithm Based on String Rewriting. In *ICAPS*, 91–98.

Gemmell, J.; Bell, G.; and Lueder, R. 2006. MyLifeBits: a personal database for everything. *ACM*, 49(1): 88–95.

Google. 2020. Apple and Google partner on COVID-19 contact tracing technology. https://tinyurl.com/442c99wm. Accessed: 2022-04-11.

Gurrin, C.; Smeaton, A. F.; Doherty, A. R.; et al. 2014. Lifelogging: Personal big data. *Foundations and Trends® in information retrieval*, 8(1): 1–125.

Hodges, S.; Williams, L.; Berry, E.; Izadi, S.; Srinivasan, J.; Butler, A.; Smyth, G.; Kapur, N.; and Wood, K. 2006. SenseCam: A retrospective memory aid. In *International conference on ubiquitous computing*, 177–193. Springer.

Hoy, M. B. 2018. Alexa, Siri, Cortana, and more: an introduction to voice assistants. *Medical reference services quarterly*, 37(1): 81–88.

Huerta, R.; and Tsimring, L. S. 2002. Contact tracing and epidemics control in social networks. *Physical Review E*, 66(5): 056115.

Jones, W. 2007. Personal information management. *Annual review of information science and technology*, 41: 453–504.

Kalnikaite, V.; Sellen, A.; Whittaker, S.; and Kirk, D. 2010. Now let me see where i was: understanding how lifelogs mediate memory. In *Human Factors in Computing*, 2045–2054.

Kalokyri, V.; Borgida, A.; and Marian, A. 2018. YourDigitalSelf: A personal digital trace integration tool. In *Proceedings of the 27th CIKM conference*, 1963–1966. ACM.

Kalokyri, V.; Borgida, A.; Marian, A.; and Vianna, D. 2017a. Integration and Exploration of Connected Personal Digital Traces. In *Proc. ExploreDB'17*, 3:1–3:6.

Kalokyri, V.; Borgida, A.; Marian, A.; and Vianna, D. 2017b. Semantic Modeling and Inference with Episodic Organization for Managing Personal Digital Traces - (Short Paper). In *Proc. ODBASE 2017*, 273–280.

Kaptelinin, V. 2003. UMEA: translating interaction histories into project contexts. In *SIGCHI conference on Human factors in computing systems*, 353–360.

Kapur, N.; Glisky, E. L.; and Wilson, B. A. 2004. External memory aids and computers in memory rehabilitation. *The essential handbook of memory disorders for clinicians*.

Kiss, I. Z.; Green, D. M.; and Kao, R. R. 2005. Disease contact tracing in random and clustered networks. *Proc. of the Royal Society B: Biological Sciences*, 1407–1414.

Li, Q.; Zheng, Y.; Xie, X.; Chen, Y.; Liu, W.; and Ma, W.-Y. 2008. Mining user similarity based on location history. In *SIGSPATIAL Advances in geographic inf. systems*, 34.

Liu, H.; and Singh, P. 2004. ConceptNet: a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4).

Meditskos, G.; Plans, P.-M.; Stavropoulos, T. G.; Benois-Pineau, J.; Buso, V.; and Kompatsiaris, I. 2018. Multi-modal activity recognition from egocentric vision, semantic enrichment and lifelogging applications for the care of dementia. *Visual Commun. and Image Representation*, 51: 169–190.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11): 39–41.

Murakami, H.; and Mitsuhashi, K. 2012. A System for Creating User's Knowledge Space from Various Information Usages to Support Human Recollection. *International Journal of Advancements in Computing Technology*, 4(22).

Naeem, U.; Bigham, J.; and Wang, J. 2007. Recognising activities of daily life using hierarchical plans. In *European Conf. on Smart Sensing and Context*, 175–189.

Organization, W. H.; for Disease Control, C.; Prevention; et al. 2015. Implementation and management of contact tracing for Ebola virus disease: emergency guideline. Technical report, World Health Organization.

Raskar, R.; Schunemann, I.; Barbar, R.; Vilcans, K.; Gray, J.; Vepakomma, P.; Kapa, S.; Nuzzo, A.; Gupta, R.; Berke, A.; et al. 2020. Apps gone rogue: Maintaining personal privacy in an epidemic. *arXiv preprint arXiv:2003.08567*.

Sap, M. 2020. ACL 2020 Commonsense Tutorial (T6). https://tinyurl.com/acl2020-commonsense. Accessed: 2022-04-11.

Schacter, D. L. 2002. *The seven sins of memory: How the mind forgets and remembers*. HMH.

Sellen, A.; and Whittaker, S. 2010. Beyond total capture: a constructive critique of lifelogging. *ACM Communic*.

Shafer, G. 1986. The combination of evidence. *International Journal of Intelligent Systems*, 1(3): 155–179.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Zhu, W. L. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM*, 1223–1237.

Stelmach. 2016. Natty Date Parser. https://github.com/joestelmach/natty. Accessed: 2022-04-11.

Thakur, S. 2016. Personalization for google now: user understanding and application to information recommendation and exploration. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 3–3.

Tulving, E. 1972. 10. Episodic and semantic memory. *Organization of memory/Eds E. Tulving, W. Donaldson, NY: Academic Press*, 381–403.

Van Den Hoven, E.; Sas, C.; and Whittaker, S. 2012. Introduction to this special issue on designing for personal memories: past, present, and future. *HCI*, 27(1-2): 1–12.

Vianna, D.; Yong, A.-M.; Xia, C.; Marian, A.; and Nguyen, T. D. 2014. A Tool for Personal Data Extraction. In *Proceedings of the 10th IIWeb Workshop*.

Von Ahn, L.; and Dabbish, L. 2008. Designing games with a purpose. *Communications of the ACM*, 51(8): 58–67.

Wanzare, L. D. A. 2020. *Script acquisition: a crowdsourcing and text mining approach*. Ph.D. thesis, Saarländ.

Wanzare, L. D. A.; Zarcone, A.; Thater, S.; and Pinkal, M. 2016. A Crowdsourced Database of Event Sequence Descriptions for the Acquisition of High-quality Script Knowledge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 3494–3501. European Language Resources Association (ELRA).

Whittaker, S.; Bellotti, V.; and Gwizdka, J. 2006. Email in personal information management. *ACM*, 49(1): 68–73.