Entity Footprinting: Modeling Contextual User States via Digital Activity Monitoring

Yousefi, Zeinab; Vuong, Thanh Tung; Al-Ghossein, Marie; Ruotsalo, Tuukka; Jacucci, Giulio ...

2024-04

Association for Computing Machinery

http://hdl.handle.net/10138/576330

Yousefi, Z, Vuong, T T, Al-Ghossein, M, Ruotsalo, T, Jacucci, G & Kaski, S 2024, 'Entity Footprinting: Modeling Contextual User States via Digital Activity Monitoring', ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 14, no. 2. https://doi.org/10.1145/3643893

Entity Footprinting: Modeling Contextual User States via Digital Activity Monitoring

ZEINAB R. YOUSEFI, Aalto University, Helsinki, Finland
TUNG VUONG, University of Helsinki, Helsinki, Finland
MARIE ALGHOSSEIN, University of Helsinki, Helsinki, Finland
TUUKKA RUOTSALO, LUT University, Lappeenranta, Finland and University of Copenhagen, Kobenhavn, Denmark
GIULIO JACCUCI, University of Helsinki, Helsinki, Finland
SAMUEL KASKI, Aalto University, Helsinki, Finland and University of Manchester, Manchester, UK

Our digital life consists of activities that are organized around tasks and exhibit different user states in the digital contexts around these activities. Previous works have shown that digital activity monitoring can be used to predict entities that users will need to perform digital tasks. There have been methods developed to automatically detect the tasks of a user. However, these studies typically support only specific applications and tasks, and relatively little research has been conducted on real-life digital activities. This article introduces user state modeling and prediction with contextual information captured as entities, recorded from real-world digital user behavior, called *entity footprinting*—a system that records users' digital activities on their screens and proactively provides useful entities across application boundaries without requiring explicit query formulation. Our methodology is to detect contextual user states using latent representations of entities occurring in digital activities. Using topic models and recurrent neural networks, the model learns the latent representation of concurrent entities and their sequential relationships. We report a field study in which the digital activities of 13 people were recorded continuously for 14 days. The model learned from this data is used to (1) predict contextual user states and (2) predict relevant entities for the detected states. The results show improved user state detection accuracy and entity prediction performance compared to static, heuristic, and basic topic models. Our findings have implications for the design of proactive recommendation systems that can implicitly infer users' contextual state by monitoring users' digital activities and proactively recommending the right information at the right time.

CCS Concepts: • **Information systems** → **Information retrieval**; • **Human-centered computing** → **User models;**

Additional Key Words and Phrases: Entity footprinting, user intent modeling, personal assistant, real-world tasks

ACM Reference Format:
Zeinab R. Yousefi, Tung Vuong, Marie AlGhossein, Tuukka Ruotsalo, Giulio Jaccuci, and Samuel Kaski. 2024. Entity Footprinting: Modeling Contextual User States via Digital Activity Monitoring. *ACM Trans. Interact. Intell. Syst.* 14, 2, Article 9 (April 2024), 27 pages. https://doi.org/10.1145/3643893

Authors' addresses: Z. R. Yousefi, Aalto University, Tietotekniikantalo, Konemiehentie 2, 02150 Espoo, Finland; e-mail: zeinab.rezaeiyousefi@aalto.fi; T. Vuong, M. AlGhossein, and G. Jaccuci, University of Helsinki, Helsinki, Finland; e-mails: vuong@cs.helsinki.fi, marie.alghossein@gmail.com, giulio.jacucci@helsinki.fi; T. Ruotsalo, LUT University, Lappeenranta, Finland and University of Copenhagen, Kobenhavn, Denmark; e-mail: tr@di.ku.dk; S. Kaski, Aalto University, Helsinki, Finland and University of Manchester, Manchester, UK; e-mail: samuel.kaski@aalto.fi.

ACM 2160-6455/2024/04-ART9
https://doi.org/10.1145/3643893

1 INTRODUCTION

Our digital life is composed of receiving, processing, communicating, and producing information. We tend to organize digital activities around tasks that are contextualized by entities, such as apps, documents, people, and various keywords. These entities are semantic data objects which have properties corresponding to real-world objects they represent [28] and specify the context of our activities. We accomplish our digital tasks by a set of interactions with entities on our digital devices, which then trigger a series of screen transitions following each interaction. The problem faced by users when engaged in digital tasks is how to allocate their limited cognitive resources to find and access the required entities from a wide range of data [51]. Digital life is characterized by multitasking and frequent interruptions, requiring frequent activity switches. For instance, consider a user engaging in multiple tasks every day; for each task, she works on different documents, opens different applications, browses the Web with specific keywords related to tasks, and communicates with colleagues about tasks. Therefore, during a day, the user is at different states working on multiple tasks that are associated with a set of entities. Furthermore, many forms of these interactions are repetitive—we visit the same websites, send messages and reply to e-mails, and open previously visited documents. With the growing number of entities, searching has become increasingly important for finding information on personal computers. This retrieval process can be time consuming and cognitively challenging, as the entity to be retrieved (e.g., a file, name of a person, or an address on the Web) may be difficult to recall. As a result, information overload can pose an additional challenge to the progression of digital tasks. Individuals have to remember which entities are associated with each task to restore the information related to that task if needed. Consequently, designing personal assistants and contextual recommendation systems that can understand the different states of the user and can support the management of tasks has gained increasing interest [12, 35, 45, 55, 71, 75].

Over the past few decades, the use of assistive technologies and tools has changed how information work is carried out. Many diverse user interfaces and interactive techniques have been developed to facilitate accessing previously used items and managing tasks [12, 26, 28, 47]. Examples of this include Web page recency lists [31, 46], showing previously opened documents [66, 73], and personal information management systems [7, 11, 49]. By exposing users to a larger variety of what may be of interest to them, they can find what they are looking for quickly and efficiently. For example, in the context of cloud-based platforms such as Google Drive and Microsoft Office 365, recommendations are intended to facilitate access to the documents users are likely to need in the near future, thereby eliminating the burden of memorizing folder structures and automating document management [74]. However, the majority of proposed personal assistants are based on heuristic methods that consider recency and frequency without modeling the user's activities.

Screen recording of digital devices (e.g., laptops, tablets, and smartphones) can provide a wealth of information about the ongoing digital tasks of the users, and consequently their states. Manual maintenance of such information collections and the analysis of data acquired from screen recordings are time consuming and not feasible for long-term studies involving weeks or months. To capture users' ongoing states, we need to develop a representation of their activities. The use of activity mining to automate maintenance of such collections appears to be a promising alternative [55]. Activity mining extracts unique activities from a stream of interactions with entities by utilizing interaction histories. The existing literature in activity mining demonstrates varying degrees of success in limited study setups in labs. Despite previous efforts, most approaches struggle to maintain two important aspects, namely (a) considering cross-app rich entities or (b) modeling the temporal behavior.

Despite a large body of work devoted to modeling user digital behavior, most of the advances were focused on predetermined interaction logs (e.g., only query logs, e-mail, or Web browsing history) [40, 68, 76], or data acquisition has been limited to a certain application or predefined tasks [30, 36, 43, 75]. In particular, the context of the user task is mainly determined based on the user's Web activity, such as recent Web queries issued by the user [15, 42] or the blogpost or Web document the user is composing [5, 17, 25, 35]. However, there are many other sources of contextual information that can be useful in determining the user state, such as 24/7 digital behavioral recordings that are not restrained to a specific application or a type of user input. Previous approaches have not been effective in utilizing rich features that are present on the user screen in real-life digital activities, as well as considering complex co-occurrence between different types of entity appearing on the screen, which this article aims to address. Furthermore, users' interests are dynamic and constantly changing, and are influenced by their previous behavior. Identifying users' dynamic preferences based on their historical behavior can be challenging but are essential for personal recommendation systems. Some of the previous studies fail to capture the sequential development of contexts over time, or only model linear dynamics of user representations, which are insufficient to capture nonlinear dynamics in human behavior [12, 28, 35, 45, 71].

There has been fairly little research on approaches that automatically learn the user states and accordingly predict the user's needs in real-life digital activities. In this article, we present *entity footprinting*, which is an approach for entity recommendation in realistic everyday digital tasks, based on user model learned from images captured from the screen. To collect the user's 24/7 digital behavioral recordings, we employed a screen monitoring approach that captured all user interaction data and generated visual content (e.g., visual content presented to the user on the screen) across application boundaries. A user state model is built of heterogeneous, multiple (temporal and topical) aspects of data that can be contextualized by several entities, such as applications, documents, people, and various keywords. The model is then utilized to predict a subsequent user state and entities relevant to that state. To this end, we aim to answer two main research questions. Our first question aims to identify the user state:

> *RQ1*: *Can we automatically identify and distinguish users' states from their everyday digital activities?*

Beyond the state identification, we are particularly interested in predicting which entities the user will be interested in working on next, which leads to our second question:

> *RQ2*: *Does user state prediction help in recommending more relevant entities to users in the context of their daily digital activities?*

The model that solves the aforementioned issues should satisfy the following four characteristics: (1) it should follow an unsupervised learning approach (i.e., the model needs no prior knowledge about the categories of activities or the labeled data); (2) due to the high dimensionality and sparsity of the extracted data from users' screens, and therefore the high computational cost of the data processing, the data must be clustered into meaningful clusters that take textual content into account and represent different states of a user; (3) it should take into account the time-varying nature of human behavior; and (4) to recommend entities to users, the predicted states must be converted into a ranking over entities. To answer those research questions and fulfill these characteristics, we present a novel approach for data-driven modeling of users' state in their daily digital activities. This model is able to predict the entities that the user is likely to find relevant given the user's interaction history.

Due to the high dimensionality and sparsity of digital behavioral data with the size of several thousands of entities occurring in the entire recording history, the user state is modeled using a topic modeling approach wherein a topic represents a user state. Statistical co-occurrence patterns

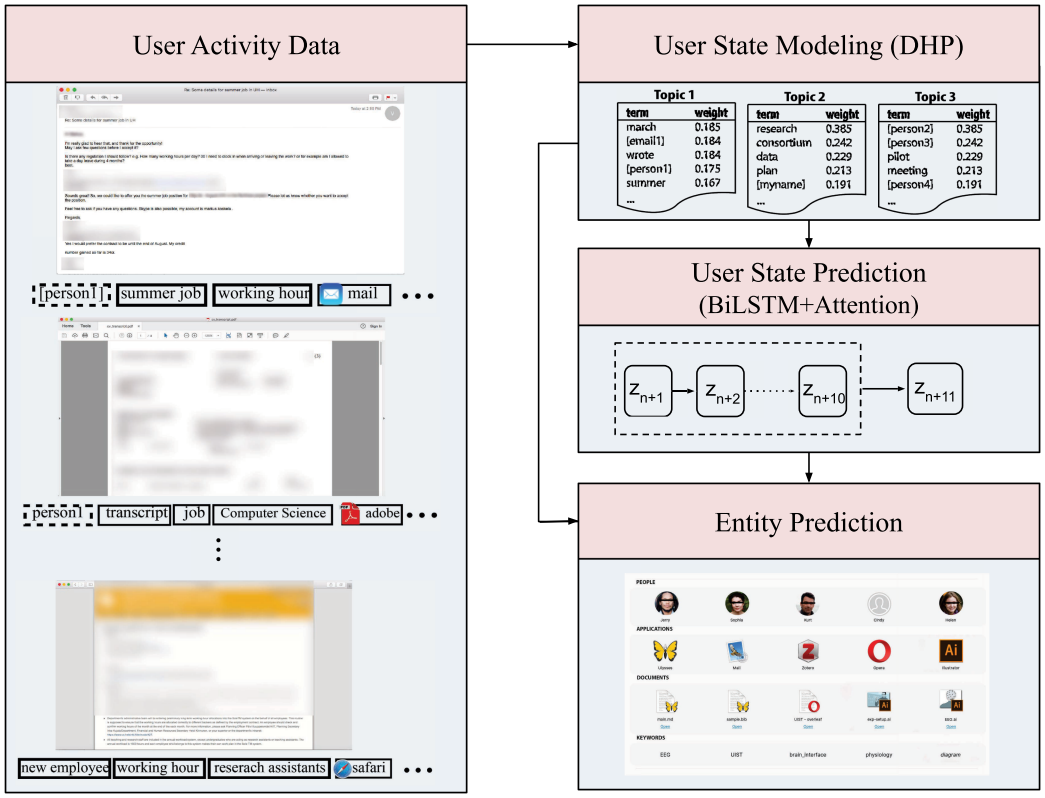

Fig. 1. Diagram of the proposed model. The model observes the user's screen, which is composed of extracted entities such as the title of the active document, the name of the active application, people's names, and keywords in that screen. In the example shown here, in the user activity data, the first two documents are an e-mail and a PDF file that the user opened, and the last document is a Web page opened in the Safari Web browser. Dimensionality is reduced by the user state modeling using the Dirichlet Hawkes Process (DHP), which takes a corpus of information objects and their time as input and converts them to topic distributions that represent different user states. Nonlinear dynamics of the topic sequence is modeled using BiLSTM and attention in user state prediction, which is responsible for the prediction of the next state that the user will attain, and entity prediction identifies the most relevant entities associated to the predicted state. Predicted entities in this setting can be recommended to the user using an interface (the bottom right figure) that displays the possible entities from which the user can choose.

among entities justify the application of the topic model in identifying the underlying latent thematic structure of the data. However, this model on its own tends to suffer from disregarding the order and not taking into account the temporal behaviors of the user. Therefore, in this work, to address this problem and to capture the sequential signals underlying users' behavior sequences, we use the powerful **Bidirectional Long Short-Term Memory (BiLSTM)** model [54] to identify the sequential relatedness of states. Moreover, we employ the self-attention mechanism [10, 69], to learn a better representation of the user's state in the behavior sequence by leveraging sequential information, to accurately predict users' subsequent state and accordingly recommend the most relevant entities related to the predicted state. The diagram of the proposed model is represented in Figure 1.

To evaluate the model, we conducted an offline analysis on a collected real-world digital activities data in which all information appearing on the screens of 13 users during a period of 14 days was captured automatically via screen monitoring and converted to texts using **Optical Character Recognition (OCR)** [28].

The main contributions of this work can be summarized as follows:

— A new representation for characterizing digital activities: the entity footprint across boundaries of applications that utilizes contexts acquired from a monitoring system to capture the user state.
— A user model capable of predicting the user state in digital life based on entity footprinting and predicting needed entities at the right time.
— An empirical evaluation showing how the proposed user model improves the prediction performance compared to baseline models.

The article is structured as follows. In the following section, a discussion of related work is provided, and in Section 3, we introduce the data acquisition approach in entity footprinting and the user interface implemented in this work. In Section 4, we present the user model overview and problem formulation including user state modeling, sequence modeling, and entity recommendation. Dataset creation and experimental exploration is presented in Section 5. We evaluate our proposed method and compare it with the baselines in Section 6. Finally, we present a discussion and conclusion in Sections 7 and 8, respectively.

2 RELATED WORK

The purpose of our work can be viewed, in broad terms, as the user state identification from contextual information collected at the entity level. A large body of research has been undertaken on user behavior modeling at the intersection of several different research topics: personal information management, digital task recognition, information retrieval, and recommendation systems. There are algorithmic approaches that aim to find how contextual information can be leveraged for document ranking and making recommendations tailored to the context of each individual user. Examples of contextual signals are pre-query sessions [73], click-through data [6], or pre-visited pages [34], or a combination of behavioral-based signals [64]. Works related to our research can broadly be grouped into research examining task-centric personal information management and context-aware recommendation systems. A comprehensive study can be found in the work of Jacucci et al. [28], where several different types of digital activity monitoring systems and related entity-based retrieval and recommendation systems are discussed.

2.1 Task-Centric Personal Information Management

One relevant area within the context of this article is research in task-centric personal information management for the support of users of digital devices. In this line of research, they assume that the user behavior is a mixture of tasks and each task is composed of a set of information items. Examples of this kind of task manager are introduced in other works [2, 19, 32, 60], in which the creation, maintenance, and collection of such information were done manually. In addition to manual support systems, there are different semiautomatic research prototypes for a task-centric support of information work [29, 57]. The clustering part was done manually and the task switches were detected automatically in these works. As another alternative, automatic detection of tasks and activities from interaction histories was introduced in several works [44, 45, 48, 52, 53, 55]. The task-specific window grouping was done based on the title of the windows in the work of Oliver et al. [45]. However, Rattenbury [48] clustered tasks by the context-aware (program names and accessed documents) pattern mining approaches. To identify task clusters, in the work of

Schmidt et al. [53], hierarchical clustering was employed based on the semantic similarity of the document contents. By applying these advancements, it has become possible to automatically split the digital work of a user into segments corresponding to different tasks. These studies either used activity properties, such as access sequence, duration, mouse clicks, and keystrokes, or content properties, such as content of documents, title of windows, and active applications. Seeliger et al. [55] combined these properties to cluster the documents visited by information workers. Satterfield et al. [52] tried to automatically associate existing task descriptions with information that users access as they work on the tasks, and consequently generate the representation of a user's work using different natural language processing techniques (TF-IDF, W2V, etc.). Except for the work of Seeliger et al. [55], evaluations in this field concentrated on lab setups with short time intervals, ranging from a few hours to a week. Although a longer period of usage (around 40 work days) was examined by Seeliger et al. [55], none of these studies model the dynamics of user behavior while interacting with the computer and instead mainly focus on clustering digital tasks and activities. Moreover, most of these approaches are tailored to a specific application or specific task (e.g., information seeking), as opposed to being either application or task agnostic. For example, the focus of Mirza et al. [44] was on classifying a user's digital activities into six specific activity types: writing, reading, communicating, Web browsing, system browsing, and miscellaneous, and Satterfield et al. [52] restricted their study to an information-seeking task for software developers. Another limitation in the recent relevant work [52] is that they assumed that the times of task switching must be indicated, so it is not a fully automatic approach. Task descriptions were also captured manually in this work, which can be impossible in realistic settings. Even in the specified tasks, it is possible that the vocabulary users use to describe their tasks does not match exactly with the words commonly found within the content of those tasks.

Task identification based on the log files has attracted the attention of the scientific community for a long time and can be served for various purposes, from time spent on tasks [9], to understanding the information needs of users [33], to support resumption of suspended or interrupted tasks. Researchers investigated the effects of explicitly representing information associated with a task on the productivity of users. Dragunov et al. [12] demonstrated that data collected from user interactions with information objects (files, e-mails, documents, contacts, etc.) could be converted into a task template that can assist users in accomplishing their tasks. These templates were used to aggregate information and associate relevant resources to each task. Information resources in this work were documents and software tools necessary to accomplish tasks. According to Brdiczka et al. [4], routine tasks are characterized by temporal regularity of user actions (e.g., switching between applications, switching windows). The tasks were labeled by users, and each task was associated with a set of documents and applications. This data was used to train a model that constructed a task representation that was based on a distribution of temporal patterns defining the user's routine. A trained model and found patterns were then used to identify the task from an unseen sequence. However, they did not aim to predict the task context or to examine more extensive sources of contextual information, but the model was trained based on predefined interaction data, such as the interaction history of specific applications. Ideally, a personal information management system would monitor digital behavior across a variety of applications to determine what the user is currently doing.

2.2 Context-Aware Recommendation Systems

In the past few decades, recommendation systems have gained popularity among researchers, and they have become a part of our daily lives due to their ability to facilitate finding relevant information in the rapidly expanding digital world. A successful and effective recommendation system depends on accurately identifying the users' intents or interests. Conventional recommendation

systems were limited to one type of data, such as to find a match for movies, books, and songs, and focused on recommending new items to the users. These recommender systems collect information from users, create and update users' profiles, and recommend information tailored to the user's profile based on similarities across users as in a collaborative filtering system. However, modern virtual personal assistants are increasingly utilizing a variety of signals derived from users' search histories to build models of their users and better predict their short-term and long-term future interests [58]. There are several commercially available recommendation systems that operate on smartphones and provide resources based on the user's current context, such as Google Now and Microsoft Cortana. In particular, Google Now attempts to model both short-term search intents as well as long-term interests and habits based on a series of search logs from a user for several months [21]. In another work, Song and Guo [61] introduced the concept of extracting patterns from search history and using those patterns to recommend information to users at specific times during the day.

Researchers have examined the use of task context not only for information management but also for recommendation systems [15, 23, 34, 56, 70]. Users' contextual information, such as search and interaction histories, spatiotemporal information, or demographics, can leverage recommendation systems to provide tailored experiences for their users. Rhodes and Maes [50] provided an example of an early proactive search setup that demonstrated the benefit of using contextual information for proactive recommendations. The context in their work is determined from the text that is written or read in a word processor, and this context is used as a query to the search system. The query runs continuously in the background and displays a list of documents that are related to the document that is currently being read or written by the user. The downside of this approach is that system logs and contextual information derived from other application sources are all ignored. Furthermore, users' historical behavior, such as their long-term history, which has been shown to improve the quality of recommendations, was not taken into account.

In the past few decades, the majority of the work on modeling context has focused on observed past user behaviors of their search history, such as query logs [14, 18] and Web browsing logs [20]. The context model was then used to update the list of initially generated recommendations, such as reranked query suggestions or automatically generated search results. For instance, the context in the work of Eickhoff et al. [14] was search engine result pages of the previous query, and the signal value was traces of user attention at the term level. Then, the candidate terms for query expansion were reranked based on the semantic correlation to those contextual terms. Letizia [40] is another introduced system that provides proactive recommendations to users during Web browsing by employing a set of heuristic rules. However, Web searches are often performed as part of a more general task [38], and therefore relying only on search history as the source of context may limit the effectiveness of recommendations. An alternative approach is to include all of the desktop data (documents stored on the computer) as the context [8] for recommendation. Initially, a set of terms closely related to the current query were identified as possible candidates. The query suggestions were then narrowed to only those terms that were semantically related to terms appearing in the desktop data.

Additionally, more recent research has incorporated data from other sources that provide a more comprehensive context. Singh et al. [59] logged user behavioral signals, such as clicks and page visits, on a real-world e-commerce site to predict the query intent of the user. Li et al. [37] considered recently read e-mails as context for recommendations. Tan et al. [63] used recently opened documents as user context. The focus of these works was on obtaining partial data, which can only be obtained through predefined applications or services. As a consequence, this would limit the use of a recommendation system. In this work, however, we integrate system logs and screen monitoring to capture all textual context as well. We focus on modeling the task context

comprehensively by considering more extensive sources from various applications, as well as considering temporal associations of the user's past interactions to predict the user's state and provide them with information relevant to the predicted state.

Most of the previous studies have focused on different neural network architectures or combinations of features, ignoring the sequential nature of user behavior sequences in real-world recommendations. These intelligent assistants focus on the recurrence of users' intent and present information that is closely related to the users' intent. Zaheer et al. [75] and Bahrainian et al. [1] addressed user modeling by predicting the topic of the user's future click and search queries, respectively. A closely related work to our proposed model is described in the work of Zaheer et al. [75], who use topic models for user modeling, using the browsing history of users. The method identifies activity patterns in a large group of users. For user modeling, the LDA topic model is used jointly with **Long Short-Term Memory (LSTM)** where activity words are URLs, and the topic model is used to discover patterns in these activities. In contrast, our work investigates the human state prediction from different types of entities extracted from the user's screen, on a large scale, and this data helps us discover individual behavior in their daily digital life. In another work, Bahrainian et al. [1] proposed a neural network based time series model with a dynamic memory that is able to learn user behavior over time and predict future search topics. In addition to predicting future topics, their proposed model can estimate the approximate time of day when a user is likely to be interested in a given search topic. Our work differs from their works in the sense of context that we apply for our modeling. We use the screen content of the user, which includes multiple (temporal and topical) aspects of the user's digital behavior. Moreover, our goal is to predict the next state of the user and accordingly recommend relevant entities to the user, not just the next app or next search query. Models that incorporate sequence models such as LSTMs (e.g., [36, 39]) have been widely employed for the modeling of interaction behavior, due to their extensive architectures and their capability to capture long-range dependencies. In recent years, the transformer model has been used to produce state-of-the-art results on a variety of tasks [69].

In a series of our work done on proactive EntityBot, Vuong et al. [71] applied latent semantic analysis with a simple bag-of-words data representation to detect users' tasks without considering the dynamic evolving between tasks. In our more recent work [28], the entity recommender system is built on a similar idea of long-term monitoring but utilizes a semisupervised machine learning approach that learns the user intent in real time. The user intent model is interactive such that it can also learn from explicit feedback from the user if it is available. The system dynamically adapts its recommendations to the most recent preferences of the user. After each interaction, the user model is updated in light of the explicit feedback provided by the user. The advantage of our approach is that we can predict the next state of the user based on their previous states, anticipating their future information needs and not just focusing on entities relevant to the user's current task.

3 DATA ACQUISITION APPROACH IN ENTITY FOOTPRINTING

Screen recording is an interesting source of data in user behavior modeling. In this section, we introduce the utilized monitoring system that continuously monitors a user's digital activities by recording screenshots of the active windows, processes, and extracts relevant information, and can generate vectors for discovering user's states from user interaction logs [71].

3.1 Digital Activity Monitoring System

The digital activity monitoring system is developed in the Mac **Operating System (OS)** and Windows OS. Both versions were implemented using the Accessibility API, a native library in

the OS. They perform identical functions that track users' digital activities and capture textual information present on the screen. The digital activity monitoring system has four modules:

— *Screen Monitoring module* captures the content in an active application window by taking screenshots at 2-second intervals and saving the screenshots as images.
— *OCR module* extracts texts from screenshots. OCR module is implemented using Tesseract 4.0,[1] which is a very accurate OCR engine.
— *OS logger* collects information about computer usage associated with the captured screenshots, including titles of active windows, names of active applications, timestamps, file paths, and URLs.
— *Keyword Extraction module* detects and extracts keywords from the OCR-processed texts. We used AllenNLP[2] to implement the Keyword Extraction system.

All OCR-processed texts and OS log information were encrypted and stored as log files on the computer. Each log entry was associated with a collection of entities, including applications, documents, keywords, and persons. Applications were names of active applications, documents were determined as titles of active windows, keywords were extracted using the Keyword Extraction module, and persons were determined by extracting senders and receivers in an e-mail (Mail, MS Outlook, and Thunderbird) and contact persons in the chat window (Skype, Messenger, WhatsApp, and Slack).

3.2 Screenshot Preprocessing

The screenshots and associated metadata, including text units produced by the OCR process as well as OS information, and entities are stored chronologically as a sequence. We merged screenshots that belong to the same information object using window titles. An information object describes the user's access to an information resource on the computer, such as a textual document, an e-mail, a folder, a file, an instant message, a Web page, and an application window with a unique title. For instance, in Figure 2, we merged screenshots that belong to an e-mail and its replies with the same subject (left pane) as a single information object, and screenshots of a Google doc with a unique title (right pane) as another information object. We focused on the content of the information object that the users read and produced by extracting only information change on the screen. For this process, we utilized a frame difference technique in which the two temporally adjacent screen frames (of a single information object) were compared, and the differences in pixel values were determined. In other words, terms that appeared in the same pixels in the two adjacent screen frames were excluded from the information object.

4 USER MODEL IN ENTITY FOOTPRINTING: PROBLEM FORMULATION

The user model in this work captures the user's interactions with the system that consists of information objects acquired from the user's screen. It then models the user's contexts, uses this model to infer the user's state at each timestep, learns the preferences of the user, and finally provides the relevant entities to the user. An illustrative example of a sequence of information objects recorded from a user's screen and the corresponding predicted entities are shown in Figure 3. We consider that each interaction with the computer can be recorded as a tuple of information objects and the time at which the interaction occurred. Each information object itself consists of a collection of entities, including the title of the document, the application to which the document belongs, and the screen content (keywords and persons). The ith interaction, Ω_i, in a sequence can be expressed

[1] https://tesseract-ocr.github.io/tessdoc/4.0-with-LSTM
[2] https://allennlp.org/

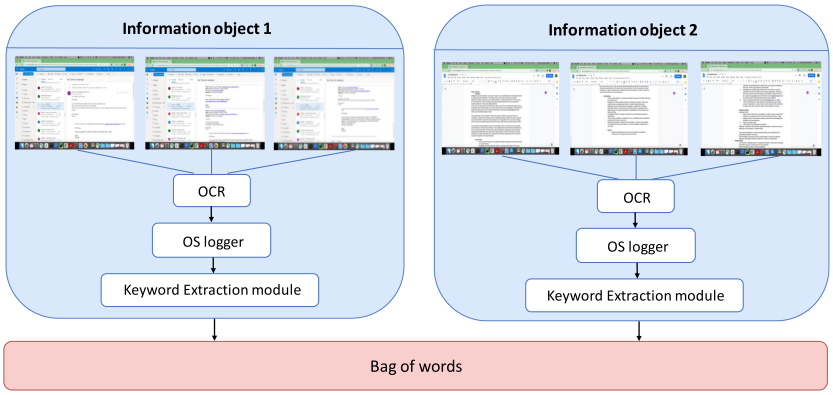

Fig. 2. Extraction of a bag of words from a digital activity monitoring system. An information object is an information resource on a computer, such as a textual document, an e-mail, a folder, a file, an instant message, a Web page, or an application window with a unique title. The illustration demonstrates how a bag of words is extracted from two information objects. The first information object contains three consecutive screenshots of e-mails with the same subject line, and the second is three consecutive screenshots of a Google doc with a unique title that has been scrolled down by a user.

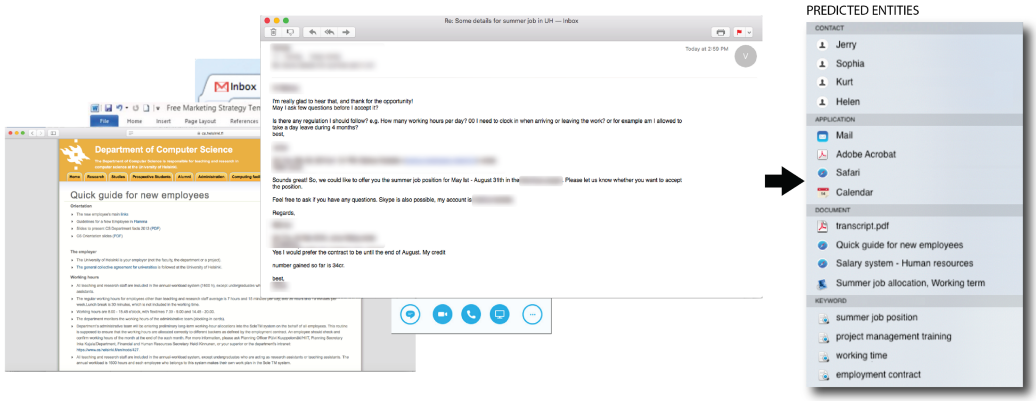

Fig. 3. Example of entity prediction. In this example, a user was engaged in human resource management on recruiting a summer trainee. The model took recently recorded screenshots (the left part) as input representing the context of the user. It then used this information to identify relevant entities and predict what the user would do next, such as read the quick guide for new employees or go through summer job applications (the right part). The model could predict the applications the user would open next and the documents they would look for. It could also suggest which people the user should contact or which keywords the user should use for searching related information.

as follows:

$$\Omega_i = (o_i, t_i), \tag{1}$$

where t_i indicates the time when an interaction with a particular information object of o_i occurred. Each o_i is

$$o_i = (s_i, a_i, w_i, p_i), \tag{2}$$

where s_i is the title of the active screen, a_i identifies the app to which the information object belongs, and w_i and p_i refer to the screen content that contains a collection of entities including

$$X = \begin{array}{c} \\ \text{entity}_1 \\ \text{entity}_2 \\ \\ \text{entity}_{|E|} \end{array} \begin{array}{c} \begin{array}{cccc} o_1 & o_2 & \dots & o_N \end{array} \\ \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \cdots & \epsilon_{1N} \\ \epsilon_{21} & \epsilon_{22} & \cdots & \epsilon_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{|E|1} & \epsilon_{|E|2} & \cdots & \epsilon_{|E|N} \end{bmatrix} \end{array}$$

Fig. 4. Input data based on the bag of words model. The element ϵ_{ij} is 1 if the ith entity ϵ_i exists in jth information object o_j. N is the number of observed information objects, E is the set of entities, and $|E|$ denotes the set's size.

keywords and people names, respectively. The sequence of user digital activities may be viewed as a series of these tuples.

$$\Omega_{1:n} = \{(o_1, t_1), (o_2, t_2), \dots, (o_n, t_n)\} \tag{3}$$

Based on the bag of words model, each information object o_i can be represented by a bag of individual entities $[\epsilon_1, \dots, \epsilon_{|E|}]^T$ in which nonzero elements are the entities present in the current information object. E is the set of all unique entities, including screen titles, app names, keywords, and people names extracted from the entire recording history, and $|E|$ denotes the set's size. The logged digital activities of the user are stored in the matrix $X \in \mathcal{R}^{|E| \times N}$ shown in Figure 4, where columns are a sequence of information objects o_is and rows are entities extracted from the user's screen. The (i, j)th element is 1 if the ith entity exists in the jth information object.

Given a sequence of previous information objects, we are interested in predicting which entities are likely to appear next, at the $(n + 1)$th step:

$$o_{n+1} = f(o_{1:n}). \tag{4}$$

To achieve the function f, we will utilize a machine learning model. Due to the fact that $|E|$ is too large (it can reach thousands within our dataset), implementing machine learning models directly on these highly sparse and large vectors would require enormous amounts of data. Therefore, we first cluster the stream of information objects into different states by inducing the textual content and exploiting the co-occurrence patterns among entities. In this way, we reduce the size of the dataset by formulating a semantic and meaningful representation for the collected entities. The second step in achieving the function f is to encode the history of a user by modeling a sequence of interactions. Finally, based on the encoding of history, we intend to design the model in such a way that recommends those entities to the user that are most likely to appear in the next timestep.

4.1 User State Modeling

Identifying the states of the user from a sequence of interactions with the computer can be seen as the task of clustering sequential data. By creating representations of a user's state, we can determine what tasks the user is focused on at any given time instant. To generate more advanced representations, which consider an entity's relevance within the context of the user's overall state, we focus on topic modeling approaches. These representations can be used to automatically match a topic to each state of the user to identify the task they are working on. The most extensively used topic model for clustering data is the **Latent Dirichlet Allocation (LDA)** model, where a finite number of topics are defined in advance [3]. Its enhanced model is the hierarchical Dirichlet process, which is the nonparametric counterpart of LDA and has an infinite number of topics [65]. These algorithms are designed to discover hidden thematic structures in a collection of documents and rely on the co-occurrence of words to make cluster inferences. In these methods, the probability of a word being assigned to a particular topic is determined by the word count of that topic.

In recent years, researchers have also been exploring the idea of clustering document streams into clusters based on the temporal sequence in which they arrive [13, 24]. The new models do not require a fixed size dataset; instead, they can be applied to a stream of documents arriving sequentially, with the number of clusters updated automatically. In this work, to deal with continuous stream of screens, we implement a **Dirichlet-Hawkes Process (DHP)**, which is a probabilistic generative model that combines the strengths of Bayesian nonparametrics as well as the Hawkes process [13]. The DHP is a continuous-time model for streaming data that allows for self-excitation. The key idea in the DHP is that the Hawkes process (one kind of temporal point process) is adopted to model the rate intensity of information objects, whereas the Dirichlet process is used to capture the state information objects' cluster relationships in which each cluster represents a state that contains information objects related to that state.

Let us show the latent state indicator by $z_{1:n}$. Given a stream of information objects $(o_i, t_i)_{i=1}^n$, the inference algorithm in the DHP is composed of two subroutines. First, it samples the latent cluster for the current information object o_n by sequential Monte Carlo and then updates the learned triggering kernels of the corresponding cluster in the progress. The DHP generates a series of samples $\theta^o_{1:n}$ corresponding to these information objects. Each state will have a distinctive value of θ^o_i. If there are K distinct values $\theta_{1:K}$ at time t_n, then $z_n \in 1, 2, \ldots, K, K+1$, where $z_n = K+1$ denotes a new state and $0 < z_n \leq K$ denotes an existing state. Let the uniform prior θ_0 be a $|E|$-dimensional vector (where $|E|$ denotes the size of unique entities set) where every element is a constant value. The posterior is decomposed as $P(z_n|o_n, t_n, rest) \sim P(o_n|z_n, rest)P(z_n|t_n, rest)$ by Dirichlet-multinomial conjugate relation. Then the likelihood $P(o_n|z_n, rest)$ is given by

$$P(o_n|z_n, rest) = \begin{cases} \frac{\Gamma(C^{z_n} + \sum_v^{|E|} \theta_0[v]) \prod_v^{|E|} \Gamma(C_v^{z_n} + C_v^{o_n} + \theta_0[v])}{\Gamma(C^{z_n} + C^{o_n} \sum_v^{|E|} \theta_0[v]) \prod_v^{|E|} \Gamma(C_v^{z_n} + \theta_0[v])} & \text{if } 0 < z_n \leq K \\ \frac{\Gamma(\sum_v^{|E|} \theta_0[v]) \prod_v^{|E|} \Gamma(C_v^{o_n} + \theta_0[v])}{\Gamma(C^{o_n} \sum_v^{|E|} \theta_0[v]) \prod_v^{|E|} \Gamma(\theta_0[v])} & \text{if } z_n = K+1. \end{cases} \tag{5}$$

Here, C^{z_n} is the entity count of cluster (state) z_n, C^{o_n} is the total entity count of information object o_n, and $C_v^{z_n}$ and $C_v^{o_n}$ are the corresponding counts of the vth entity. Finally, $P(z_n|t_n, rest)$ is the prior given by the DHP as

$$P(z_n|t_n, rest) = \begin{cases} \frac{\lambda_{\theta_k}(t_n)}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i^o}(t_n, t_i)} & 0 < k \leq K \\ \frac{\lambda_0}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i^o}(t_n, t_i)} & k = K+1, \end{cases} \tag{6}$$

where λ_0 is the base intensity of a background Poisson process, λ_{θ_k} is the intensity of the Hawkes process corresponding to the kth state, and $\gamma_{\theta_i^o}(t_n, t_i) = \exp(-|t_n - t_i|)$. Using these probabilities, sequential Monte Carlo sampling is used to infer the state label of each information object.

This model is able to learn a representation of each observed input at each timestep and provide an appropriate framework for generating a representation of the user state based on the observation of digital activity. Additionally, as the digital activity of the user arrives at streaming fashion and has time information, we can leverage both the content and time information to better cluster the activities, or what we call *states,* of the user. However, this model still tends to suffer from disregarding the order and not taking into account the sequential information of states and recurrent activities of the user.

4.2 User State Prediction

The state representation explained in the previous subsection is aimed to cluster what is on the user screen at each time frame. We also can compress what happens over time. Our focus is on a frequently encountered question: can we predict the kind of activity a user will undertake in the

future based on the sequence of activities observed in the past? How do past states affect the occurrence of future states? To correctly understand user preferences, one must be able to account for the information about the sequential behaviors and inherent dynamics in the behavior. Therefore, the second component of our model is sequence learning on the user state. This module is aimed to process the sequence of input and predict the most likely future continuation of the sequence, which is the state that is expected to be reached by the user. By modeling the sequences, we can learn the digital activity patterns of the users. As an example, the occurrence of one event related to checking Twitter may result in a series of events about other social media such as Facebook. Generally, when a user is working, information objects that appear in close proximity to one another tend to share a similar topic. This implies that the appearance of a specific topic is likely to be followed by the emergence of related topics in a nearby time frame.

One typical approach to model temporal dynamics in user behaviors is to use a latent autoregressive model. This algorithm updates the latent state using $h_{n+1} = f(h_n, o_n)$, and the observable state is derived from $o_{n+1} = g(h_{n+1}, o_n)$ for some data o_n. Functions f and g are nonlinear functions that can be learned from data and are commonly referred to as **Recurrent Neural Networks (RNNs)** in deep learning. One of the most used variants of RNNs is LSTM, which contains specially designed units to avoid vanishing gradients.

Our technique is based on the idea of seeing the state as a nonlinear function of the state's history and parameterizing it using an RNN. In this model, user state history can be encoded into a compact vector representation, from which the subsequent state of the user can be predicted. Encoding of user interaction history into a compact vector (representing the user's preferences) can be done using the basic paradigm of the left-to-right sequential model. Despite their popularity and efficacy, such unidirectional left-to-right models are insufficient for learning appropriate representations of user behavior sequences. These models were initially developed for types of sequential data that have natural order, such as text and time series data. Therefore, encoding is done only on data from previous items. However, users' behaviors in real-world applications may not always follow this rigidly ordered sequence [27, 62, 72]. When modeling user behavior sequences, we can consider context from both directions. LSTMs with bidirectional properties can learn input sequences both forward and backward, leading to both interpretations being concatenated and embedded within the hidden state. Our intuition behind using the BiLSTM neural network is to use all available information and effectively model the local dependencies between certain states of the user in a temporal manner.

Formally, given a state z_n at the timestep n, corresponding hidden state h_n can be derived by using the equations defining the various gates used in LSTM as follows:

$$\begin{aligned} i_n &= \sigma(z_n U^i + h_{n-1} W^i) \\ f_n &= \sigma(z_n U^f + h_{n-1} W^f) \\ o_n &= \sigma(z_n U^o + h_{n-1} W^o) \\ g_n &= \tanh(z_n U^g + h_{n-1} W^g) \\ c_n &= \sigma(f_n \odot c_{n-1} + i_t \odot g_n) \\ h_n &= o_n \odot \tanh(c_n), \end{aligned} \tag{7}$$

where c_n denotes the cell state. To capture the long-term dependencies, the LSTM cell adds internal gating mechanism. i, f, and o are the input, forget, and output gates, respectively, in Equation (7). These gates control how information is added to or removed from cell states along the sequence of state updates. z_n and h_n are the one-hot vector of input state and the LSTM hidden state at timestep n, respectively.

We divide a sequence of user states $z_1, z_2, \ldots, z_n$ into a fixed-sized sliding window of size W for $n = 1, \ldots, N$, and each sequence is formed as $\{z_{n-W+1}, \ldots, z_{n-1}, z_n\}$. Given the last W of user states in this window, the LSTM network performs the following:

$$h_n = f(h_{n-1}, z_n), \quad n = 1, \ldots, N. \tag{8}$$

The forward layer output sequence, $\overrightarrow{h}$, is iteratively calculated using inputs in a positive sequence from time $N - W$ to time $N - 1$, whereas the backward layer output sequence, $\overleftarrow{h}$, is calculated using the reversed inputs from time $N - W$ to time $N - 1$. The desired output that is the prediction of the next topic is then produced at each timestep from h_n:

$$\hat{z}_{n+1} = g(\overrightarrow{h_n}, \overleftarrow{h_n}), \tag{9}$$

where g is an arbitrary differentiable function followed by a softmax. The BiLSTM network accepts a sequence of z (state) as input and outputs the next z.

Although BiLSTM utilizes the user's sequential behavior to capture the long-term dependency in the contextual user state, this approach cannot focus on the important information and the user's main purpose within the obtained contextual state. In real-life digital activity, there are situations where a user is working on a specific topic but accidentally opens a document or clicks on a wrong link that opens an irrelevant Web page. While these actions are part of the user's behavior sequence, they are not the primary focus of the user at that time. As a result, it is crucial to contemplate the main goal of the user in each session in addition to the sequential behavior. By concentrating on the important aspects of the contextual state, we can boost the accuracy of our prediction. The attention mechanism can highlight important information by setting different weights. BiLSTM combined with the attention mechanism can enhance the prediction accuracy even further [41].

To train the model using back-propagation, the loss$(\hat{z}_{n+1}, z_{n+1})$ is measured using categorical cross entropy. The trained network can then serve as a model to predict the future state in the test dataset. The output of the network depends not only on the latest state but also on a sequence of states.

4.3 Entity Recommendation

The topic model in the first subsection provides the probability of entities at each state. The sequence model then predicts the probability of each state at the next timestep after the final softmax. By knowing these probability values at timestep n, the probability of a given entity ϵ_n assuming Z states is computed by

$$p(\epsilon_n) = \sum_{\zeta=1}^{Z} p(\epsilon_n | z_n = \zeta) p(z_n = \zeta). \tag{10}$$

Top k entities are generated by sorting entities in descending order. In other words, entities in each type (apps, documents, people, and keywords) that are most consistent with the future state are retrieved.

5 EXPERIMENTAL STUDY

To investigate the research questions, we collected data from 13 users as they accomplished their daily digital tasks. Five males and eight females with an average age of 25 years were recruited to take part in the study. Participants with higher educational backgrounds were chosen, as they were likely to use their personal laptops for work-related tasks, allowing us to collect more realistic data. Upon joining, the study participants were informed of their privacy and told that their data would

Table 1. Summary of the Collected Data from 13 Participants

Participants	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
# applications	188	173	181	83	131	181	135	71	115	234	49	149	135
# documents	1034	1299	1243	564	553	1320	606	460	532	983	359	855	810
# people	62	48	6	14	156	25	293	453	646	346	241	429	419
# keywords	44450	69416	45948	51797	22651	45299	16794	18811	29257	26992	13148	27947	47700
# information objects	5460	3483	5543	3306	1461	3787	2043	2008	2417	2096	1090	2884	2171

An information object describes an access to an information resource on the computer, such as a textual document, an e-mail, or a Web page. Entities are applications, documents, people, and keywords that are extracted from the information object.

be encrypted and stored on a secure server, and used only for research purposes. As compensation for participating, they received 120 euros.

The research was carried out in accordance with the ethical guidelines of the University of Helsinki. Regarding the data usage policy and procedure, participants were asked to complete a consent form. The research plan and informed consent form were approved by the Ethical Committee of the University of Helsinki. It is important to note that all logs are stored locally, the logging tool does not upload any data to the cloud, and all evaluation scripts utilizing these logs were run locally on the computers of participants.

The monitoring system was installed on participants' laptops, and digital activities were continuously recorded in the background thread for 14 days. The system was set to launch automatically whenever the laptop was turned on. Participants could stop the system anytime; however, we advised them to avoid doing so during the monitoring period unless it was necessary.

5.1 Data Description

The data were preprocessed into a standardized format, consisting of a stream of information objects, each comprising of the merged screenshots of documents with the same window title, which includes a set of entities: an application name, a document title, keywords and non-keyword terms in OCR-processed text units. We used frame difference methods to exclude duplicate keywords and terms constantly appearing on the screen and focused only on the information change. Due to a large number of occurrences with respect to various browsers, we decided to extract the domain names of the Web pages visited and considered them to be separate applications.

5.2 Data Analysis

Table 1 summarizes the data collected during the 2-week digital activity monitoring of 13 participants. The number of recorded information objects per participant was 2,903 (SD = 1,388), which corresponded to an average of 78 hours (SD = 73) of computer usage per participant. The average number of unique documents and unique applications accessed per participant was 811 (SD = 326) and 140 (SD = 52), respectively. An average of 241 (SD = 208) people entities were found from the data. Keyword extraction from OCR-processed text units resulted in 35,400 (SD = 16,611) keywords and 17,534 (SD = 6,859) non-keyword terms per participant.

5.3 Training Details

Hyperparameters for the DHP include setting $\lambda_0 = 0.05$ and $\gamma_0 = 0.1$. For the inference, we used sequential Monte Carlo sampling with eight particles. For BiLSTM, we used a sequence length of $W = 10$. The BiLSTM network was modeled using two layers and 64 neurons on each layer. The network parameters were learned using mini-batch stochastic gradient descent algorithm, where the batch size was set to 32, the dropout rate set to 0.5, and the learning rate initialized to 0.001.

Categorical cross entropy was used as the loss function. The loss on the validation set was also used as the criterion for the early stopping of the training. We split each user's data into training and test sets. We selected 80% of the data for training and used the remaining 20% as the test set, and the evaluation objective for prediction experiments is that given 80% data for training, we want to assess the predictive quality for individual user states and entities issued during the remaining 20% of data. We also sampled 20% of the training set as a hold-out validation set. BiLSTM models were trained for 1,000 epochs (i.e., 1,000 iterations over the entire training set) and then evaluated against the validation set. The model parameters with the best performance on the validation set were selected and then evaluated on the test set. The BiLSTM networks were implemented using TensorFlow library and trained on a machine equipped with GPUs.[3]

6 EVALUATION

The predictive performance of the model is evaluated by measuring the following:

(1) how accurately we can predict the user's upcoming states (RQ1) and
(2) how well we can predict the entities the user will use going forward (RQ2).

6.1 Performance Measures

The following standard measures were used:

— User state prediction *Accuracy* (*Acc*), which indicates the cumulative accuracy of the correctly predicted users' states (or dominant topics in information objects) over the test set for each user:

$$Acc = \frac{1}{|N_u|} \sum_u Acc_u, \tag{11}$$

where $|N_u|$ is the total number of users and Acc_u is the state prediction balanced accuracy for each user. To obtain the ground truth for evaluating state prediction accuracy, we considered the topics obtained from the DHP topic model at each timestep as the ground truth states. Due to the imbalanced dataset (which means that some states have many more samples than that of other states), we used balanced accuracy for the multiclass problem [22].

— The metric *Hitrate@k* accounts for the presence of a single correct entity within the top k predictions. However, considering that in the people and keywords category, multiple correct entities may appear on the screen simultaneously, this measure was not considered appropriate for assessing the performance of keywords and people predictions.

— *Recall@k* of document, application, people, and keyword prediction. Recall@k is computed as the average fraction of the actual used entities appearing on a participant's screen in the next steps that is successfully predicted. How many relevant entities are retrieved?

— *Precision@k* is the fraction of relevant entities within the top k predicted entities. How many retrieved entities are relevant?

— *MRR@k* (Mean Reciprocal Rank) is a statistical measure used to evaluate the rank position of the first relevant entity. MRR is the average of reciprocal rank across all test data. For example,

$$MRR = \frac{1}{|N_u|} \sum_u MRR_u = \frac{1}{|N_u|} \sum_u \frac{1}{|D_u|} \sum_{i \in D_u} \frac{1}{rank_i}, \tag{12}$$

[3] https://www.tensorflow.org/

where $|D_u|$ is the length of the test set for each user, and $rank_i$ represents the ranking of the model with the top recommended entity at the ith test point. If a rank is greater than k, the reciprocal rank is set to zero.

When evaluating hitrate, precision, recall, and MRR, we considered entities present at the next information object as ground truth entities at each time instant.

6.2 Baselines

We compare the proposed algorithm with existing algorithms that have been adapted for predicting future states and entities based on previous users' behavior:

— **Most Recently Used (MRU)** that recommends entities from most recent to least recent ones. Recency is a powerful heuristic that has been continuously applied in commercial products, such as recently used apps or recently made calls [77]. MRU predicts the next entities to be recommended according to the entities present in the last screenshot. This baseline is competitive when the user is working on activity for a long time.

— **Most Frequently Used (MFU)** that evaluates frequencies of entities and recommends them in decreasing order. Frequency is another common heuristic used in the recommendation of items and entities. The method predicts the element that will be used by the user based on how frequently it has been used previously.

— **Combined Recency and Frequency (CRF)** [16] considers all previous accesses to an entity. Equation (13) is used to calculate the entity's weighting ω_f. n is the number of previous accesses, t is the current timestep, and t_i represents the timestep where access i took place (time is defined in terms of discrete events). In our setting, $p = 2$ and $\lambda = 0.1$ resulted in the best performance.

$$\omega_f = \sum_{i=1}^{n} \frac{1}{p}^{\lambda(t-t_i)} \tag{13}$$

— *MRU_topic* uses the most recent state as the prediction—that is, it removes the sequence modeling part as it only assumes the next user activity is related to the current state of the user. MRU_topic is used to evaluate the significance of the sequential information contained in the user behavior. This baseline only makes use of the context without considering the temporal information.

These methods treat human actions passively rather than acknowledging the dynamics of the user behavior.

— *Markov chain* involves exploiting the sequential nature of user behavior and translating sessions into Markov processes:

$$P(X_{n+1} = x | X_n = x_n) = \frac{|x_n \rightarrow x|}{|x_n|}, \tag{14}$$

where $|x_n|$ represents the number of previous occurrences of state x_n and $|x_n \rightarrow x|$ represents the number of previous transitions from state x_n to x. X_n indicates the state at time n. Given the most recent access x_n, the calculated probabilities produce a ranking.

— *LDA* uses LDA topic modeling rather than the DHP in our proposed model. LDA calculates the probability of each topic (or state) at each time instant rather than clustering the dataset into states. We fed the sequence of these topic vectors into BiLSTM as input. A drawback of LDA is that the number of topics must be predetermined. By using the number of topics acquired from the DHP model, we were able to compare the state prediction accuracy of LDA with that of our proposed model.

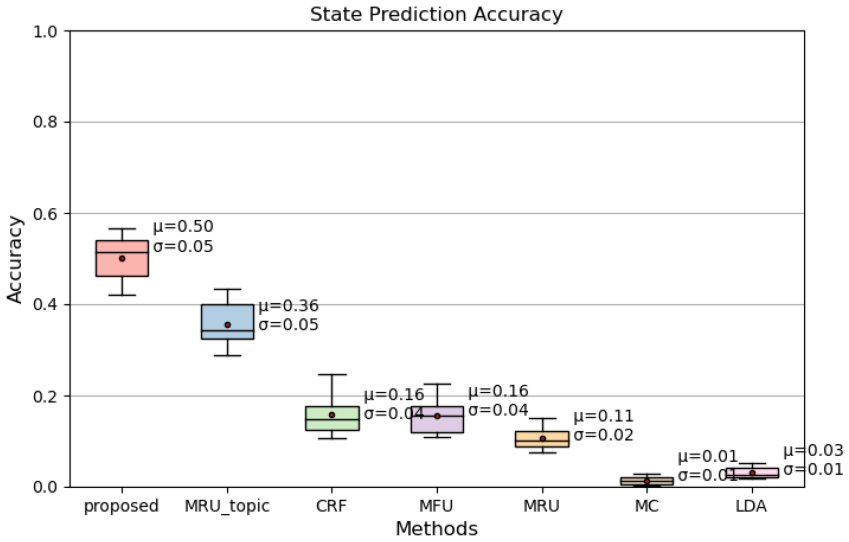

Fig. 5. State prediction accuracy in different methods. μ and σ are the average and standard deviation of the accuracy in 13 users.

6.3 Statistical Testing Procedure

To determine whether there is a statistically significant difference in performance among our approach and the baselines, two-tailed paired sample t-tests with the p-value threshold of 0.05 ($p < 0.05$) were used. To test the significance levels, we used accuracy in the state prediction, and hitrate, recall, precision, and MRR in the entity prediction as dependent variables and models as independent variables. The Shapiro-Wilk test was also used for all normality tests.

6.4 State Prediction Accuracy

To answer RQ1, we compared the predictive performance of user state for different approaches. DHP topic modeling provides us with two important outputs: ground truth states at each timestep and the distribution of entities at each state ($p(\epsilon|z)$). Prediction of the states for the baseline models are as follows. The MRU_topic model only considers the most recent state as prediction, so it does not consider the temporal dynamics of the user behavior. In other words, LSTM is not included in the modeling and only the DHP provides us with states. Heuristic baselines (CRF, MFU, and MRU) provide weight vector for all extracted entities at each timestep. To convert these weight vectors to the analogous states of the DHP, we used normalized weight vectors at each model and compared it using cosine similarity against the vector of entity distribution for each state ($p(\epsilon|z)$), and took the argmax as the predicted state. The cosine similarity between two vectors is determined by the angle between those vectors projected in a multidimensional space. As the angle decreases, the cosine similarity increases. In other words, the closer these vectors of entity distributions are to each other, the more similar the states. In the LDA-BiLSTM model, we took the argmax of the predicted topic vector as the state at each timestep. Figure 5 shows the comparison between our proposed model and baselines. By leveraging sequential dynamics between different user states, our proposed approach improves over the static model MRU_topic, Markov chain, and other baseline models. There was a significant difference in the state prediction accuracy of the proposed method compared to baselines. Paired t-test p-values were at $p < 0.0001$. This result suggests that modeling the temporal dynamics of the user behavior plays an important role in predicting the

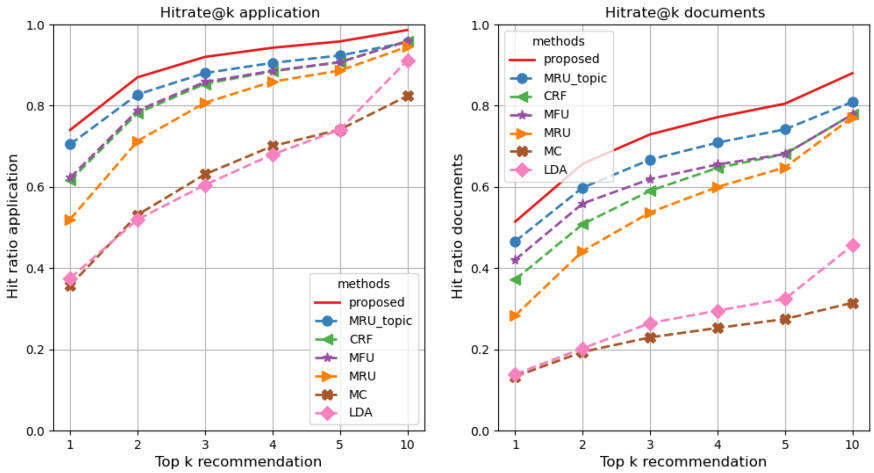

Fig. 6. Total hit ratio @ 1, 2, 3, 4, 5, and 10 of applications (left) and documents (right).

next contextual state of the user. The advantage of LSTMs over hidden Markov models based on Markovian assumptions and have a finite number of hidden states is that they have a continuous space memory that enables them to make predictions based on longer-term observations. As we mentioned earlier, one drawback of the LDA topic model is that it relies on determining the number of topics in advance. This is challenging for this type of data log. Moreover, documents are assumed to be exchangeable in the LDA model [3]. This assumption is too restrictive when addressing streaming documents, since documents about the same topic are not exchangeable as topics evolve over time [67]. Document collections in our study exhibit evolving content.

6.5 Entities Prediction

Since at each time instant there is only one application (a_i) and one document (title of the screen s_i), described at the beginning of Section 4, it is possible to compare the hitrate@k of these two types of entities across different models. Figure 6 shows the performances of models in terms of hitrate@k for applications and documents (left and right, respectively). Our approach consistently outperforms the baselines over the top 1 to 5 and 10 recommendation. These improvements are statistically significant, as tested with a paired Student t-test, $p < 0.01$. Our proposed approach achieves a high hitrate of more than 0.5 even with a single document prediction (hitrate@1) and more than 0.7 even with a single app prediction. The results indicate that our approach was successful in anticipating the applications and documents that the users actually would use in the unseen data.

Precision@20 of different models in predicting four different types of entity, including application, document, people, and keywords, is shown in Figure 7. Precision means the percentage of prediction results that are relevant. It can be seen that the precision in document and application and keyword prediction is significantly different ($p < 0.05$) compared to all baselines except MRU and MC in the application. The difference of precision@20 for the people recommendation is significant only in MRU and MC.

Figure 8 shows performances of the the models in terms of Recall@20, which is an indicator of how well the model is able to preload the next entity between the top 20 predicted entities. The results of the application and document prediction indicate that recall of our approach outperforms the baselines, and the difference was significant ($p = 0.05$), demonstrating the importance of

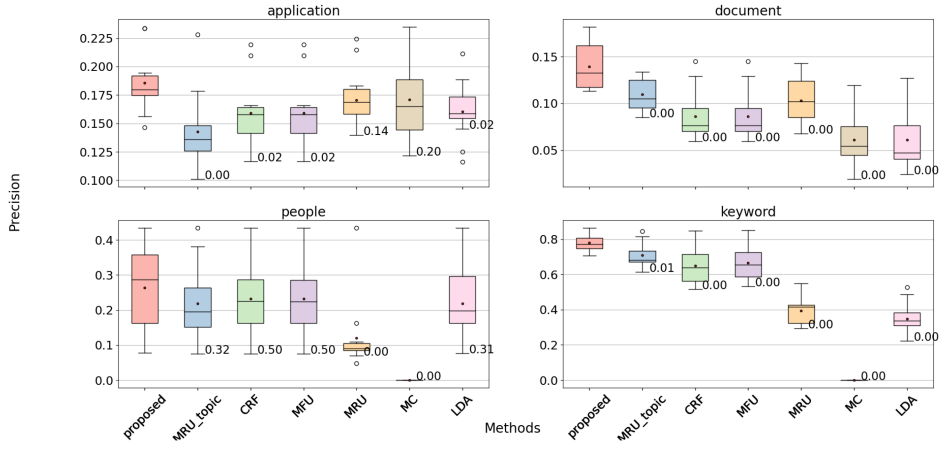

Fig. 7. Total Precision@20 of entities (application, document, people, and keyword) in different methods. Numbers written on the lower cap are paired t-test p-values for the proposed method and baselines, (significant at $p < 0.05$).

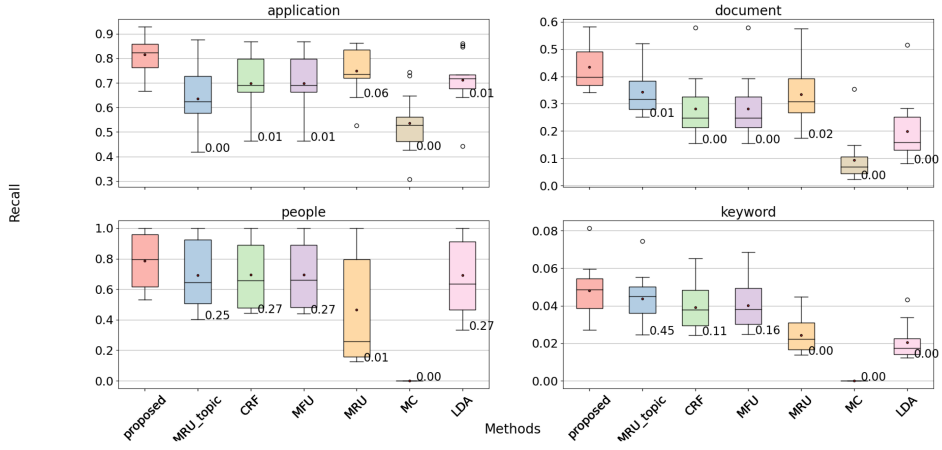

Fig. 8. Total Recall@20 of entities (application, document, people, and keyword) in different methods. Numbers written on the lower cap are paired t-test p-values for the proposed method and baselines (significant at $p < 0.05$).

modeling user states. However, no significant difference was found in the recall of people and keyword prediction between the models.

Figure 9 shows the MRR for evaluating the effectiveness of improved prediction when it is applied to the entity that is ranked highest. This metric calculates the average or mean of the inverse of the ranks at which the first relevant entity was retrieved. As can be seen from the results, our proposed method performs better, particularly in the document or application, where the user wants to select one entity to click on and the proposed method provides more reliable predictions.

In summary, our model outperformed the baselines in almost all measures since the model explicitly considers the temporal behavior of the user, whereas static and heuristic baselines simply output the ranking of the entities without any sequence information. It is true that the LDA-BiLSTM model considers the sequential nature of the user's behavior and it models the dependencies

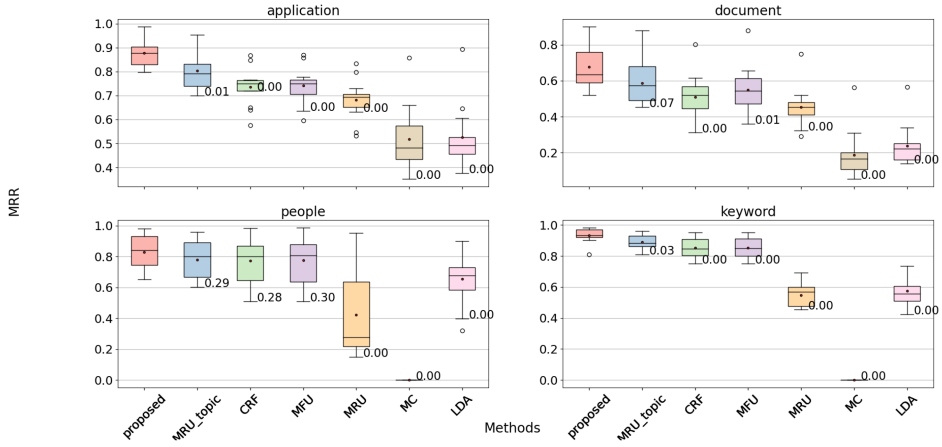

Fig. 9. Total MRR of entities (application, document, people, and keyword) in different methods. Numbers written on the lower cap are paired t-test p-values for the proposed method and baselines (significant at $p < 0.05$).

between entities within a lower-dimensional space; however, the disadvantage of the LDA topic model is that it needs a predefined number of topics, which are suitable for static datasets of long documents. Yet, in our study, we are dealing with streaming corpora that sometimes contain information objects with a limited number of entities, which makes the inference of latent topic distributions more challenging. Modeling user states makes a major contribution to the entity prediction. This is visible in the results that the more correctly predicted user states are, the higher the hitrate, precision, recall, and MRR.

7 DISCUSSION

The main contribution of this article is to introduce entity footprinting for predicting user information needs using contextual information available on the screen of the user. This system collects an individual's entity footprints from personal digital devices and automatically extracts state representation that can be used to learn semantic relationships of the collected information and to find related entities such as documents, apps, people, and keywords. A user study provides evidence that our system is able to proactively produce relevant resources and is suited for re-finding previously seen entities.

It is important to note that entity footprinting differs significantly from other recommendation tasks, such as movies, songs, and shopping items. In this research, users have a lot of information about the entities with which they have previously interacted, and they have a clear objective with regard to finding or relocating specific entities (e.g., documents, apps, people) when they work with their digital devices. Therefore, a successful recommendation system in the setting of entity footprinting requires an accurate recommendation algorithm. A more accurate recommendation indicates that the proposed method can reduce users' manual search effort by providing them with more relevant and useful information.

Furthermore, the entity footprinting presented in this article differs from the other personal information management systems in three ways. First, our approach in this work is on being proactive, in the sense that it does not require any action from users and instead exploits context from users' screens and past interactions to predict users' needs in the future and provide them with information that is relevant to their predicted tasks. Aside from that, users' everyday digital

activities are heterogeneous, meaning that they are not limited to a specific application and can switch across several applications. Therefore, the second difference is that the entity footprinting is principally based on screen recordings, hence making it a general system that is agnostic to tasks users perform or applications they use for their tasks. In this work, we did not conduct any experimental study with the controlled settings in the lab, and we examined in-the-wild data collection and real-world tasks. By using a single data source (users' screen), we were able to create a rich user model without requiring any human supervision. Entity footprinting was examined especially on datasets acquired from mostly knowledge workers; however, it was not limited to knowledge work and can be applied to other types of computer users as well. Participants in our user study were engaged in different types of tasks ranging from writing a thesis and coding to checking social media, online shopping, and reading news. Within these tasks, participants took part in different activities, and their intents and preferences changed frequently. Due to this drift in intents and preferences for entities over time, entity footprinting should be time sensitive. Therefore, the third difference is that our proposed model explicitly takes into account the temporal behavior of the user.

The proposed system can augment the human with a digital memory of entities interacted. This memory can be used in different applications building on personal digital data. Among these applications are (1) *proactive search*, in which users are provided with information based on their past behavior rather than explicitly querying for information, and are able to learn their interests and search preferences based on their history; (2) *time line search*, which aids in recalling events and searching for specific information by displaying the information on a graphical time line; and (3) *associative recall*, which deals with specific relationships between entities. It is possible that we remember some partial information, but not the exact information we are looking for. Entity footprinting provides cues that can facilitate associative recall.

Results on state prediction accuracy showed that the proposed method is able to capture the users' rapidly evolving preferences and consequently provide them with entities that are actually used in their tasks. Our findings provide evidence that considering contextual as well as temporal information can help the entity footprinting system identify significantly more relevant entities than other baselines when performing real-world digital tasks. Nevertheless, utilizing a digital activity monitoring method is not without limitations. Here, we acknowledge the limitations of our study and outline potential research areas for future studies.

Artifact Access. Some information, such as Web bookmarks, may always be visible on the active windows regardless of the task at hand. The model may be confused by this information.

Experiment Limitations. Our findings were based on a 13-person experiment that lasted for 2 weeks. To ensure that our findings will be valid for a broader population, a larger experiment over a longer period of time will be necessary. Although our observations have provided us with valuable insights, the possibility of improving the prediction accuracy could have been enhanced by longer sessions and more data.

Generalization. It is not feasible to generalize from one person's collection of personal information objects to another because of the abundance of specialized tasks, keywords, and entities used. Therefore, our model should operate at an individual level by processing data from each user's device, without relying on collective patterns across multiple users.

Privacy. The monitoring system introduced in this work may contain sensitive information. However, this is a common issue with most personal assistant systems. Some participants disabled the monitoring temporarily during some activities. More study of the concealed data could assist in automating the process of setting the privacy boundaries that users expect.

Influence on User Behavior. An evaluation of performance could be conducted, rather than focusing on relevance, to quantify the usefulness and impact that comprehensive entity footprinting can have on users' daily digital activities.

7.1 Future Work

There are several future directions for this work. One could improve the model by incorporating other temporal features such as duration and timing of events and activities, and here, the Hawkes process can play a crucial role in creating more accurate models. There is important information to be gained by analyzing the precise interval between two events to understand the dynamics of the underlying behavior. The characteristics of these data establish a fundamental difference from independent and identically distributed time series data, where time is viewed as an index rather than randomly distributed variables. The second direction of extensions can also further investigate the dependence between the user states and their transitions. Most user states discovered by our models generally correspond to repetitive human tasks. By recognizing which routine a user probably engages in, a collection of related entities can be recommended.

The data stored in entity footprinting is in textual form. Even those files that contain images, videos, or voices are stored and retrieved by the name of the file in textual form. It is, however, possible to convert these types of data into textual data (e.g., speech-to-text, visual concept detection) and augment them with extracted textual information.

Furthermore, there is no comparison with other temporal dynamics modelings (e.g., RNN, transformers). Using RNN at the entity level may result in higher accuracy in predicting entities, as it can take advantage of entity-level statistics. However, the aim of this work was to investigate the possibility of modeling user states via digital activity monitoring. Therefore, comparing other advanced models such as transformer-based models and the proposed method is an interesting area of future work.

8 CONCLUSION

Despite the fact that entity recommendation systems are becoming a common feature of personal assistants and commercial platforms, research in this area is limited to specific applications or predefined tasks. However, our focus in this article was to study the applicability of entity footprinting in everyday digital life. We investigated how much the proposed approach is able to understand the users' states while they perform their everyday digital tasks using heterogeneous applications. This has been enabled by a digital activity monitoring system, which allowed context extraction across application boundaries. By automatically predicting and presenting relevant information in advance, users can easily access the information without having to formulate specific queries. The proposed model (1) is unsupervised and does not need any knowledge about the categories of activities or tasks, (2) clusters the high-dimensional digital activity data into a meaningful states, (3) considers the time-varying nature of the human behavior by a sequential and attention model, and (4) represents the predicted states as ranking over entities to be able to recommend top-ranked entities.

To validate our approach, we implemented it in the introduced entity footprinting system and conducted a user study with a realistic dataset. We investigated the impact of the user's state dynamic evolution on finding more relevant entities. In the earlier study based on the EntityBot system, the user model relied on linear modeling to address the challenges inherent in high dimensionality, limited explicit interaction, and being real time for interactive use. However, in this work, we developed a more complex model using the DHP topic model and BiLSTM network to learn the user states and their dynamic behavior. In this work, we presented a prediction framework for user state that incorporates two factors influencing user digital daily behavior: context and user preferences. We evaluated this framework with a 2-week, 13-subject field trial and compared it to heuristic, static, and other baselines.

ACKNOWLEDGMENTS

This research was made possible with support from multiple sources. We received co-funding from the Research Council of Finland through the MyModel 357270 project. Additionally, this was partially supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI and decision numbers: 359853, 352915, 350323), and the Horizon 2020 FET program of the European Union (grant CHIST-ERA-20-BCI-001), and EU H2020 project CO-ADAPT.

REFERENCES

[1] Seyed Ali Bahrainian, Fattane Zarrinkalam, Ida Mele, and Fabio Crestani. 2019. Predicting the topic of your next query for just-in-time IR. In *Proceedings of the European Conference on Information Retrieval*. 261–275. https://doi.org/10.1007/978-3-030-15712-8_17

[2] Victoria Bellotti and J. D. Thornton. 2006. Managing activities with TVACTA: TaskVista and activity-centered task assistant. In *Proceedings of the SIGIR Workshop on Personal Information Management*. 8–11.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (Jan. 2003), 993–1022.

[4] Oliver Brdiczka, Norman Makoto Su, and James Bo Begole. 2010. Temporal task footprinting: Identifying routine tasks by their temporal patterns. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*. 281–284. https://doi.org/10.1145/1719970.1720011

[5] Jay Budzik and Kristian J. Hammond. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*. 44–51. https://doi.org/10.1145/325737.325776

[6] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 875–883. https://doi.org/10.1145/1401890.1401995

[7] Sergey Chernov, Gianluca Demartini, Eelco Herder, Michał Kopycki, and Wolfgang Nejdl. 2008. Evaluating personal information management using an activity logs enriched desktop dataset. In *Proceedings of the 3rd Personal Information Management Workshop (PIM'08)*, Vol. 155.

[8] Paul-Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. 2007. Personalized query expansion for the Web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 7–14. https://doi.org/10.1145/1277741.1277746

[9] Irina D. Coman. 2007. An analysis of developers' tasks using low-level, automatically collected data. In *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering*. 579–582. https://doi.org/10.1145/1287624.1287715

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018). https://doi.org/10.48550/arXiv.1810.04805

[11] Xin Luna Dong and Alon Halevy. 2005. A platform for personal information management and integration. In *Proceedings of VLDB 2005 PhD Workshop*. 26.

[12] Anton N. Dragunov, Thomas G. Dietterich, Kevin Johnsrude, Matthew McLaughlin, Lida Li, and Jonathan L. Herlocker. 2005. TaskTracer: A desktop environment to support multi-tasking knowledge workers. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI'05)*. ACM, 75–82. https://doi.org/10.1145/1040830.1040855

[13] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J. Smola, and Le Song. 2015. Dirichlet-Hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 219–228. https://doi.org/10.1145/2783258.2783411

[14] Carsten Eickhoff, Sebastian Dungs, and Vu Tran. 2015. An eye-tracking study of query reformulation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 13–22. https://doi.org/10.1145/2766462.2767703

[15] Henry Feild and James Allan. 2013. Task-aware query recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 83–92. https://doi.org/10.1145/2484028.2484069

[16] Stephen Fitchett and Andy Cockburn. 2012. AccessRank: Predicting what users will do next. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2239–2242. https://doi.org/10.1145/2207676.2208380

[17] Ang Gao and Derek Bridge. 2009. Using shallow natural language processing in a just-in-time information retrieval assistant for bloggers. In *Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science*. 103–113. https://doi.org/10.1007/978-3-642-17080-5_13

[18] Jianfeng Gao and Jian-Yun Nie. 2012. Towards concept-based translation models using search logs for query expansion. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 1–10. https://doi.org/10.1145/2396761.2530275

[19] Victor M. González and Gloria Mark. 2004. "Constant, constant, multi-tasking craziness" managing multiple working spheres. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 113–120. https://doi.org/10.1145/985692.985707

[20] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. 2015. Context- and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 383–392. https://doi.org/10.1145/2766462.2767709

[21] Ramanathan Guha, Vineet Gupta, Vivek Raghunathan, and Ramakrishnan Srikant. 2015. User modeling for a personal assistant. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*. 275–284. https://doi.org/10.1145/2684822.2685309

[22] Isabelle Guyon, Kristin Bennett, Gavin Cawley, Hugo Jair Escalante, Sergio Escalera, Tin Kam Ho, Núria Macià, Bisakha Ray, Mehreen Saeed, Alexander Statnikov, and Evelyne Viegas 2015. Design of the 2015 ChaLearn AutoML challenge. In *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN'15)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2015.7280767

[23] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016), 9–27. https://doi.org/10.1016/j.eswa.2016.02.013

[24] Xinran He, Theodoros Rekatsinas, James Foulds, Lise Getoor, and Yan Liu. 2015. HawkesTopic: A joint model for network inference and topic modeling from text-based cascades. In *Proceedings of the International Conference on Machine Learning*. 871–880. https://proceedings.mlr.press/v37/he15.html

[25] Monika Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin. 2005. Query-free news search. *World Wide Web* 8, 2 (2005), 101–126. https://doi.org/10.1145/775152.775154

[26] Donghan Hu and Sang Won Lee. 2020. ScreenTrack: Using a visual history of a computer screen to retrieve documents and Web pages. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13. https://doi.org/10.1145/3313831.3376753

[27] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying personalized recommendation with user-session context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 1858–1864.

[28] Giulio Jacucci, Pedram Daee, Tung Vuong, Salvatore Andolina, Khalil Klouche, Mats Sjöberg, Tuukka Ruotsalo, and Samuel Kaski. 2021. Entity recommendation for everyday digital tasks. *ACM Transactions on Computer-Human Interaction* 28, 5 (2021), 1–41. https://doi.org/10.1145/3458919

[29] Steven Jeuris, Steven Houben, and Jakob Bardram. 2014. Laevo: A temporal desktop interface for integrated knowledge work. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 679–688. https://doi.org/10.1145/2642918.2647391

[30] Paul Jones, Shivani Sharma, Changsung Moon, and Nagiza F. Samatova. 2017. A network-fusion guided dashboard interface for task-centric document curation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI'17)*. ACM, 481–491. https://doi.org/10.1145/3025171.3025177

[31] Shaun Kaasten and Saul Greenberg. 2001. Integrating back, history and bookmarks in Web browsers. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*. 379–380. https://doi.org/10.1145/634067.634291

[32] Victor Kaptelinin. 2003. UMEA: Translating interaction histories into project contexts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 353–360. https://doi.org/10.1145/642611.642673

[33] Andrew J. Ko, Robert DeLine, and Gina Venolia. 2007. Information needs in collocated software development teams. In *Proceedings of the 29th International Conference on Software Engineering (ICSE'07)*. IEEE, 344–353. https://doi.org/10.1109/ICSE.2007.45

[34] Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. 2015. Predicting search intent based on pre-search context. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 503–512. https://doi.org/10.1145/2766462.2767757

[35] Markus Koskela, Petri Luukkonen, Tuukka Ruotsalo, Mats Sjöberg, and Patrik Floréen. 2018. Proactive information retrieval by capturing search intent from primary task context. *ACM Transactions on Interactive Intelligent Systems* 8, 3 (2018), 1–25. https://doi.org/10.1145/3150975

[36] Seokjun Lee, Rhan Ha, and Hojung Cha. 2018. Click sequence prediction in Android mobile applications. *IEEE Transactions on Human-Machine Systems* 49, 3 (2018), 278–289. https://doi.org/10.1109/THMS.2018.2868806

[37] Cheng Li, Mingyang Zhang, Michael Bendersky, Hongbo Deng, Donald Metzler, and Marc Najork. 2019. Multi-view embedding-based synonyms for email search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 575–584. https://doi.org/10.1145/3331184.3331250

[38] Yuelin Li and Nicholas J. Belkin. 2008. A faceted approach to conceptualizing tasks in information seeking. *Information Processing & Management* 44, 6 (2008), 1822–1837. https://doi.org/10.1016/j.ipm.2008.07.005

[39] Yang Li, Samy Bengio, and Gilles Bailly. 2018. Predicting human performance in vertical menu selection using deep learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–7. https://doi.org/10.1145/3173574.3173603

[40] Henry Lieberman et al. 1995. Letizia: An agent that assists web browsing. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. 924–929.

[41] Gang Liu and Jiabao Guo. 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* 337 (2019), 325–338. https://doi.org/10.1016/j.neucom.2019.01.078

[42] Jiqun Liu, Shawon Sarkar, and Chirag Shah. 2020. Identifying and predicting the states of complex search tasks. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*. 193–202. https://doi.org/10.1145/3343413.3377976

[43] Jiaxin Mao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Constructing click models for mobile search. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 775–784. https://doi.org/10.1145/3209978.3210060

[44] Hamid Turab Mirza, Ling Chen, Ibrar Hussain, Abdul Majid, and Gencai Chen. 2015. A study on automatic classification of users' desktop interactions. *Cybernetics and Systems* 46, 5 (2015), 320–341. https://doi.org/10.1080/01969722.2015.1012372

[45] Nuria Oliver, Greg Smith, Chintan Thakkar, and Arun C. Surendran. 2006. SWISH: Semantic analysis of window titles and switching history. In *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI'06)*. ACM, 194–201. https://doi.org/10.1145/1111449.1111492

[46] George Papadakis, Ricardo Kawase, Eelco Herder, and Wolfgang Nejdl. 2015. Methods for Web revisitation prediction: Survey and experimentation. *User Modeling and User-Adapted Interaction* 25, 4 (2015), 331–369. https://doi.org/10.1007/s11257-015-9161-7

[47] Florian Pellegrin, Zeynep Yücel, Akito Monden, and Pattara Leelaprute. 2021. Task estimation for software company employees based on computer interaction logs. *Empirical Software Engineering* 26, 5 (2021), 1–48. https://doi.org/10.1007/s10664-021-10006-4

[48] Tye Lawrence Rattenbury. 2008. *An Activity Based Approach to Context-Aware Computing*. Ph.D. Dissertation. University of California, Berkeley.

[49] Tim Reynaert. 2015. Re-finding Physical Documents: It is like a dream come true. Master's Thesis. Vrije Universiteit Brussel.

[50] Bradley James Rhodes and Pattie Maes. 2000. Just-in-time information retrieval agents. *IBM Systems Journal* 39, 3.4 (2000), 685–704. https://doi.org/10.1147/sj.393.0685

[51] Claudia Roda. 2011. *Human Attention in Digital Environments*. Cambridge University Press.

[52] Chris Satterfield, Thomas Fritz, and Gail C. Murphy. 2020. Identifying and describing information seeking tasks. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*. 797–808. https://doi.org/10.1145/3324884.3416537

[53] Benedikt Schmidt, Johannes Kastl, Todor Stoitsev, and Max Mühlhäuser. 2011. Hierarchical task instance mining in interaction histories. In *Proceedings of the 29th ACM International Conference on Design of Communication*. 99–106. https://doi.org/10.1145/2038476.2038495

[54] Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681. https://doi.org/10.1109/78.650093

[55] Alexander Seeliger, Benedikt Schmidt, Immanuel Schweizer, and Max Mühlhäuser. 2016. What belongs together comes together: Activity-centric document clustering for information work. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. 60–70. https://doi.org/10.1145/2856767.2856777

[56] Chirag Shah. 2018. Information fostering-being proactive with information seeking and retrieval: Perspective paper. In *Proceedings of the 2018 Conference on Human Information Interaction and Retrieval*. 62–71. https://doi.org/10.1145/3176349.3176389

[57] Jianqiang Shen, Lida Li, Thomas G. Dietterich, and Jonathan L. Herlocker. 2006. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*. 86–92. https://doi.org/10.1145/1111449.1111473

[58] Milad Shokouhi and Qi Guo. 2015. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 695–704. https://doi.org/10.1145/2766462.2767705

[59] Gyanit Singh, Nish Parikh, and Neel Sundaresan. 2012. Rewriting null e-commerce queries to recommend products. In *Proceedings of the 21st International Conference on World Wide Web*. 73–82. https://doi.org/10.1145/2187980.2187989

[60] Greg Smith, Patrick Baudisch, George Robertson, Mary Czerwinski, Brian Meyers, Daniel Robbins, and Donna Andrews. 2003. GroupBar: The TaskBar evolved. In *Proceedings of the 2003 Conference for the Computer-Human Interaction Special Interest Group of the Human Factors Society of Australia (OZCHI'03)*.

[61] Yang Song and Qi Guo. 2016. Query-Less: Predicting task repetition for NextGen proactive search and recommendation engines. In *Proceedings of the 25th International Conference on World Wide Web*. 543–553. https://doi.org/10.1145/2872427.2883020

[62] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450. https://doi.org/10.1145/3357384.3357895

[63] Jiwei Tan, Xiaojun Wan, Hui Liu, and Jianguo Xiao. 2018. QuoteRec: Toward quote recommendation for writing. *ACM Transactions on Information Systems* 36, 3 (2018), 1–36. https://doi.org/10.1145/3183370

[64] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2010. Potential for personalization. *ACM Transactions on Computer-Human Interaction* 17, 1 (2010), 1–31. https://doi.org/10.1145/1721831.1721835

[65] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101, 476 (2006), 1566–1581.

[66] Tuan A. Tran, Sven Schwarz, Claudia Niederée, Heiko Maus, and Nattiya Kanhabua. 2016. The forgotten needle in my collections: Task-aware ranking of documents in semantic information space. In *Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval*. 13–22. https://doi.org/10.1145/2854946.2854971

[67] J. W. Uys, N. D. Du Preez, and E. W. Uys. 2008. Leveraging unstructured information using topic modelling. In *Proceedings of the 2008 Portland International Conference on Management of Engineering and Technology (PICMET'08)*. IEEE, 955–961. https://doi.org/10.1109/PICMET.2008.4599703

[68] Christophe Van Gysel, Bhaskar Mitra, Matteo Venanzi, Roy Rosemarin, Grzegorz Kukla, Piotr Grudzien, and Nicola Cancedda. 2017. Reply with: Proactive recommendation of email attachments. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 327–336. https://doi.org/10.1145/3132847.3132979

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[70] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. 2012. Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies* 5, 4 (2012), 318–335. https://doi.org/10.1109/TLT.2012.11

[71] Tung Vuong, Giulio Jacucci, and Tuukka Ruotsalo. 2017. Watching inside the screen: Digital activity monitoring for task recognition and proactive information retrieval. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–23. https://doi.org/10.1145/3130974

[72] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32. https://doi.org/10.1609/aaai.v32i1.11851

[73] Ryen W. White, Paul N. Bennett, and Susan T. Dumais. 2010. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. 1009–1018. https://doi.org/10.1145/1871437.1871565

[74] Xuhai Xu, Ahmed Hassan Awadallah, Susan T. Dumais, Farheen Omar, Bogdan Popp, Robert Rounthwaite, and Farnaz Jahanbakhsh. 2020. Understanding user behavior for document recommendation. In *Proceedings of the Web Conference 2020*. 3012–3018. https://doi.org/10.1145/3366423.3380071

[75] Manzil Zaheer, Amr Ahmed, and Alexander J. Smola. 2017. Latent LSTM allocation joint clustering and non-linear dynamic modeling of sequential data. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 3967–3976. https://proceedings.mlr.press/v70/zaheer17a.html

[76] Qian Zhao, Paul N. Bennett, Adam Fourney, Anne Loomis Thompson, Shane Williams, Adam D. Troy, and Susan T. Dumais. 2018. Calendar-aware proactive email recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 655–664. https://doi.org/10.1145/3209978.3210001

[77] Xin Zhou and Yang Li. 2021. Large-scale modeling of mobile user click behaviors using deep learning. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 473–483. https://doi.org/10.1145/3460231.3474264

Received 5 July 2022; revised 25 April 2023; accepted 2 January 2024