EntityBot: Actionable Entity Recommendations for Everyday Digital Task

Tung Vuong*
University of Helsinki
Helsinki, Finland
vuong@cs.helsinki.fi

Salvatore Andolina*
University of Palermo
Palermo, Italy
salvatore.andolina@unipa.it

Giulio Jacucci*
University of Helsinki
Helsinki, Finland
giulio.jacucci@helsinki.fi

Pedram Daee*
Aalto University
Helsinki, Finland
pedram.daee@aalto.fi

Khalil Klouche
University of Helsinki
Helsinki, Finland
khalil.klouche@helsinki.fi

Mats Sjöberg
CSC – IT Center for Science
Espoo, Finland
mats.sjoberg@csc.fi

Tuukka Ruotsalo
University of Helsinki
Helsinki, Finland
University of Copenhagen
Copenhagen, Denmark
tuukka.ruotsalo@helsinki.fi

Samuel Kaski
Aalto University
Helsinki, Finland
University of Manchester
Manchester, UK
samuel.kaski@aalto.fi

ABSTRACT

Our everyday digital tasks require access to information from a wide range of applications and systems. Although traditional search systems can help find information, they usually operate within one application (e.g., email client or web browser) and require the user's cognitive effort and attention to formulate proper search queries. In this paper, we demonstrate EntityBot, a system that proactively provides useful and supporting entities across application boundaries without requiring explicit query formulation. Our methodology is to exploit the context from screen frames captured every 2 seconds to recommend relevant entities for the current task. Recommendations are not restricted to only documents but include various kinds of entities, such as applications, documents, contact persons, and keywords representing the tasks. Recommendations are actionable, that is, a user can perform actions on recommended entities, such as opening documents and applications. The EntityBot also includes support for interactivity, allowing the user to affect the recommendations by providing explicit feedback on the entities. The usefulness of entity recommendations and their impact on user behavior has been evaluated in a user study based on real-world tasks. Quantitative and qualitative results suggest that the system had an actual impact on the tasks and led to high user satisfaction.

CCS CONCEPTS

• **Information systems**; • **Human-centered computing**;

*Tung Vuong, Salvatore Andolina, Giulio Jacucci, and Pedram Daee contributed equally.

CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9156-6/22/04.
https://doi.org/10.1145/3491101.3519910

KEYWORDS

Proactive information retrieval, real-world tasks, user intent modeling

ACM Reference Format:
Tung Vuong, Salvatore Andolina, Giulio Jacucci, Pedram Daee, Khalil Klouche, Mats Sjöberg, Tuukka Ruotsalo, and Samuel Kaski. 2022. EntityBot: Actionable Entity Recommendations for Everyday Digital Task. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3491101.3519910

1 INTRODUCTION

Our everyday digital tasks are composed of processing, producing, and communicating information. The information that we manage is often encoded into many different formats: documents, emails, instant messages, web pages, software codes, contact persons, and keywords. The considerable amount of information that we process while the task requires our focus, concentration, and memory makes our everyday digital life extremely challenging. Searching is generally effective for finding and accessing information. However, despite the continuous advancement of search systems, including context-aware query expansion and auto-complete [16, 17], the query formulation phase remains a cognitively intensive activity that requires the user's effort and attention, interrupting the main task [5].

Recommender systems, on the other hand, better assist the users in their daily routines [8, 9]. Recommender systems are increasingly becoming an integral element of our everyday digital tasks by continuously monitoring, updating user models, and recommending information that may be useful without requiring the user's explicit actions [3, 4, 14]. Researchers have studied recommendations offered to users contextually while they are performing specific tasks [1, 2, 6, 13]. However, research in this area has been focused on evaluating recommender systems in the lab using specific tasks predetermined by the researchers, which does not answer whether such systems effectively support users in real-world situations. Also, current recommender systems typically provide information within one application (e.g., emails or web pages), whereas everyday digital tasks require access to information from various applications and systems. Moreover, while complex search tasks are interactive, most recommender systems typically do not include support for interactivity, such as allowing users to provide feedback to improve the recommendations.

In [10] we introduce the EntityBot, a recommender system that targets both cross-app input and cross-app rich entities. We aim to design a system that would make the user's everyday digital tasks less challenging and more productive by recommending the right information to use at the right time. EntityBot builds an unsupervised model of a user's topical activities from screen monitoring and digital activity monitoring data. Subsequently, the model is used to predict the user's context from unseen user activity and automatically retrieve relevant entities such as people, applications, documents, or keywords (see video illustration[1]). For the design of EntityBot, we focused on the following three design principles:

- *Entity-centric approach.* We present an approach for entity recommendation in everyday digital tasks based on learning from continuous image streams of the screen frames. With EntityBot, users are recommended with applications, documents, persons, and keywords based on their current tasks; therefore, recommendations are not limited to only documents but various kinds of entities that could be used to represent the task. EntityBot also makes recommendations more actionable by including hyperlinks/shortcuts that permit direct access to documents and applications. Because EntityBot synthesizes all information entities in a commonplace, it can reduce the overhead cost in manual switching between applications to search, locate, and access the necessary resources (e.g., documents and/or software applications).
- *Task-related context.* We design the EntityBot with the capability to access task-related information across application boundaries. Unlike prior work with access to partial data, which is only obtainable through predefined applications or services [9, 12] (for instance, data may include page visits [12], click-through data [9], or pre-search context [11] or a combination of behavioral signals[15]), EntityBot was designed to function without the need for specific application-dependent customization. The solution we employ uses digital activity monitoring to extract the texts shown on people's computer screens. We use this rich context source to extract relevant entities such as documents, applications, people, and keywords.
- *Interactive feedback and learning.* We designed the EntityBot to allow users to affect the recommendation through interactive relevance feedback so that it could improve the overall quality of recommendations. While EntityBot can work properly with continuous screen frame data alone, the user interface design also includes easy mechanisms to provide explicit relevance feedback on recommended entities.

[1] https://youtu.be/8laoTM-3jbc

To understand the effectiveness and usefulness of recommended entities in everyday digital tasks, we conducted a study using participants' real-world data and tasks. EntityBot was installed on users' laptops for two weeks to collect screen frames and digital activities and conduct unsupervised learning of the entities' representation and their relationship during actual work tasks. After this, users participated in an experimental session resuming previous work tasks. EntityBot is set up to recommend entities visualized on the user interface during users' work. Users may open recommended documents, applications, and contacts or give feedback by selecting an entity that, in turn, performs an update on the model, leading to a new set of recommended entities. The study received ethical approval from the University of Helsinki in Finland. Our results and findings can be found in [10].

2 DEMONSTRATION DESCRIPTION

The demonstration will be based on a predefined user model trained with interactional data of one user who consented to share it for the purpose of a system demo. The data is semi-anonymous. All person entities were masked with artificial names. During the demonstration, conference participants on the site will receive an explanation of one of the tasks (which is about planning for participation in the CHI conference) and will be invited to resume the task. In the case of an online system demo, participants will remotely control the presenter's computer to work with the task. As the participants start performing actions on the computer, such as opening the website of the CHI conference or searching for a hotel near the conference venue, the system will respond by showing relevant entity recommendations in real-time.

3 ENTITYBOT COMPONENTS

Here, we describe the main components of the system demo. The system consists of three main components. Screen monitoring and digital activity monitoring systems extract entities across application boundaries. An online machine learning method learns about user interests in real-time based on screen-monitoring data and, if available, explicit feedback (source code on github[2]). A user interface (UI) presents the list of recommended entities (ref. Figure 1).

3.1 Screen Monitoring and Digital Activity Monitoring

The screen monitoring and digital activity monitoring systems are composed of four main modules. The first module is a screen monitoring *(SM) system* that records screen frames of active windows every 2 seconds. SM was implemented into two versions: a Mac OSX version and an MS Windows OS version. We used the Accessibility API, a native OS library, to develop both versions. The API in two versions performs an identical function that saves the screen frames of active windows as images. The second module is an optical character recognition *(OCR) system* that detects and extracts text from the screen frames. We utilized Tesseract 4.0 [3] for the OCR system. The third module is Entity Extraction *(EE) system* detects and extracts available person names and keyword entities from the OCR-processed screen frames. We utilized the IBM

[2] https://github.com/HIIT/Entity-Recommendation-for-Everyday-Digital-Tasks
[3] https://github.com/tesseract-ocr/tesseract/wiki

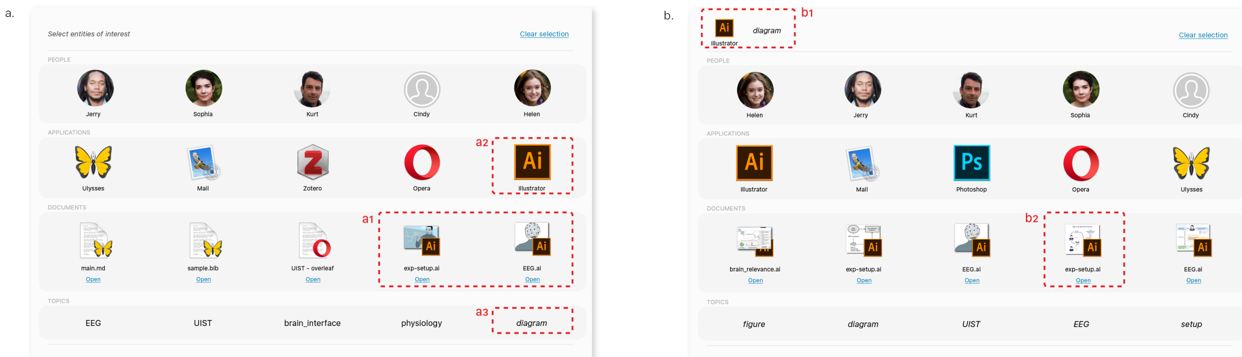

Figure 1: Two states of EntityBot's user interface [10]. Recommended entities are displayed within four rows, each with five items: people, applications, documents, and keywords. The user can select entities of interest by clicking on them, which updates the recommendations. Example: In (a), the user sees entities related to her current work. She notices figures she has made for one of her papers (a1). She clicks on "Illustrator" (an application for editing vector graphics) (a2), then on the topic "diagram" (a3). (b) As a result, the entities of interest are displayed in the top area (b1), and the system updates the recommendations accordingly with the user's selection. In the documents row, she selects an illustration (b2) that she will modify for use in her new paper.

Bluemix Natural Language Understanding API [4] to extract those entities. The last module is *OS logger* that collects OS information associated with the screen frames. The OS information includes names of active applications, titles of active windows, available URLs of web pages, or available file paths of files/documents stored on the computer's local hard drive. In addition, the OS logger also collects timestamps of when the screen frames are taken.

3.2 Modeling and Recommendation

The modeling approach receives screen monitoring data and digital activity monitoring data and prepares recommendations to show on the system's user interface. The user may give explicit feedback on the entities, and the system updates the recommendation accordingly. In our solution, the relationship between entities is established mainly based on their co-occurrence and partially through temporal closeness. We define a context at each time step as a vector. Contexts in our setting include OCR-processed screen frames, recorded OS information, and extracted entities. Our approach is inspired by the bag-of-words model in which a context is represented as a bag of individual entities: applications, keywords, named entities, and non-entity terms. The logged contexts are stored in the matrix $X \in \mathbb{R}^{|E|\times|C|}$, where the element (i, j) describes the tf-idf weighting of the entity i in context j, E and C are the sets of entities and observed contexts. Because the number of context vectors is large, containing thousands of screen frames and entities, we compress the matrix X into lower-dimensional spaces, such that co-occurring entities should get similar representations. In order to do so, we perform truncated singular value decomposition (truncated SVD) on X to get the projection matrix $W_K \in R^{|C|\times K}$ that enables us to project entities into a latent K-dimensional space. We defined the user interest as a linear model in this latent space,

$$r^E = XW_K\theta, \tag{1}$$

[4]https://www.ibm.com/watson/services/natural-language-understanding/

where $r^E \in \mathbb{R}^{|E|}$ is the vector containing relevance of all entities (we use r_i^E to refer to the i^{th} element) and θ is the K-dimensional latent user interest.

We follow the keywords-documents connection idea in [7] to connect the interest to the relevance of contexts by assuming that the relevance of a context is a weighted sum of the relevance of entities that have appeared in it as:

$$r_j^C = \sum_{i=1}^{|E|} p_{(i|j)} r_i^E, \tag{2}$$

where r_j^C refers to the relevance of the j^{th} context (with some abuse of notation), and $p_{(i|j)}$ is the likelihood of the i^{th} entity being present in the j^{th} context. This likelihood is not available, but it can be approximated based on the logged contexts (i.e., X). We normalize the columns of X so that elements of each context vector sum up to one and denote the resulting matrix as $\hat{X}$. Using this approximation and writing Equation 2 in a vector format gives $r^C = \hat{X}^\top r^E$. Finally, by using Equation 1 we can directly connect the user interest to contexts

$$r^C = \hat{X}^\top XW_K\theta. \tag{3}$$

In the online phase of the study, the most recent recorded screen frames were considered as "contexts" and were fed to the model for the prediction of user interest. Explicit feedback provided by the user on the entities through the user interface is used to update the recommendation. This feedback is connected through the shared user interest θ. By assuming a Multivariate Gaussian prior on θ, we can complete the Bayesian inference loop and compute the posterior of θ after receiving explicit feedback and recent contexts. The posterior has a closed-form solution and is employed to estimate the expected relevance of entities and contexts (Equations 1 and 3). Respectively, the relevance estimation is used to rank all entities (of different types: people, keywords, and applications) and contexts (with their corresponding linked documents) to be recommended to the user.

3.3 User Interface

Figure 1 illustrates EntityBot's UI. It implements three specific features: 1) showing the entities being recommended by the system, 2) allowing the selection of entities of interest by the user (explicit feedback), and 3) allowing direct action on entities when relevant. In the following, we describe how each feature was implemented in our experimental setup.

3.3.1 Showing the entities being recommended by the system. The interface is shown in Figure 1. Entities are displayed within four rows, each with five items representing an entity type: people, applications, documents, and keywords. Each document has a hyperlink below the icon for quick access to the resources associated with the current task. Upon clicking on the hyperlink, it activates a script that opens the resource by the application associated with it; for instance, it opens the document using the file's path, the URL of the webpage, and the unique ID of the email. The recommended entities are ranked horizontally from left to right in each row. Since the main purpose is to show a small variety of the most relevant entities, the ranking is not visually emphasized. As users perform their tasks, the system progressively updates the recommendations and the changes are reflected on the UI.

3.3.2 Allowing the selection of entities of interest by the user (explicit feedback). Users can provide explicit feedback by selecting the recommended entities on the UI with a click. Accordingly, the selected entities or entities of interest appear in the area at the top and the recommendations in every row are updated. The model considers both recent contexts and the selection of entities for making recommendations. More entities can then be selected and added to the entities of interest at the top of the screen, providing an explicit way to influence the recommendations. Entities of interest can be removed from the selection by clicking the cross that appears at their upper-right-hand corner when the mouse cursor hovers over their icon. Removal of an entity of interest from the selection sends neutral feedback on the selected entity to the system, which updates the recommendations accordingly. The entire selection of entities of interest can be reset by clicking the "Clear selection" button on the right.

3.3.3 Allowing direct action on entities when relevant. An important feature of the system is that it makes the recommendations actionable. While work on translating recommended people and keywords into potential actions is ongoing, the present version simply allows one to open recommended applications and documents directly.

4 CONCLUSIONS

In this demo, we introduce EntityBot [10] to the CHI community. EntityBot assists the user in everyday digital tasks by inferring the context of the task and providing them with the right information to use at the right time through interactive entity recommendations[5].

REFERENCES

[1] Salvatore Andolina, Khalil Klouche, Tuukka Ruotsalo, Patrik Floréen, and Giulio Jacucci. 2018. Querytogether: Enabling entity-centric exploration in multi-device collaborative search. *Information Processing & Management* 54, 6 (2018), 1182–1202.

[2] Salvatore Andolina, Valeria Orso, Hendrik Schneider, Khalil Klouche, Tuukka Ruotsalo, Luciano Gamberini, and Giulio Jacucci. 2018. Investigating Proactive Search Support in Conversations. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 1295–1307. https://doi.org/10.1145/3196709.3196734

[3] Salvatore Andolina, Valeria Orso, Hendrik Schneider, Khalil Klouche, Tuukka Ruotsalo, Luciano Gamberini, and Giulio Jacucci. 2018. SearchBot: Supporting Voice Conversations with Proactive Search. In *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Jersey City, NJ, USA) *(CSCW '18)*. Association for Computing Machinery, New York, NY, USA, 9–12. https://doi.org/10.1145/3272973.3272990

[4] Apple. 2021. About Siri Suggestions on iPhone. https://support.apple.com/en-gb/guide/iphone/iph6f94af287/ios.

[5] Richard Boardman and M. Angela Sasse. 2004. "Stuff Goes into the Computer and Doesn't Come out": A Cross-Tool Study of Personal Information Management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 583–590. https://doi.org/10.1145/985692.985766

[6] Jay Budzik, Kristian J. Hammond, and Larry Birnbaum. 2001. Information access in context. *Knowledge-Based Systems* 14, 1–2 (2001), 37–53. https://doi.org/10.1016/S0950-7051(00)00105-2

[7] Pedram Daee, Joel Pyykkö, Dorota Glowacka, and Samuel Kaski. 2016. Interactive Intent Modeling from Multiple Feedback Domains. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. ACM, New York, NY, USA, 71–75. https://doi.org/10.1145/2856767.2856803

[8] 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* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2239–2242. https://doi.org/10.1145/2207676.2208380

[9] Stephen Fitchett, Andy Cockburn, and Carl Gutwin. 2014. Finder Highlights: Field Evaluation and Design of an Augmented File Browser. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3685–3694. https://doi.org/10.1145/2556288.2557014

[10] 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 Trans. Comput.-Hum. Interact.* 28, 5, Article 29 (aug 2021), 41 pages. https://doi.org/10.1145/3458919

[11] 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* (Santiago, Chile) *(SIGIR '15)*. Association for Computing Machinery, New York, NY, USA, 503–512. https://doi.org/10.1145/2766462.2767757

[12] 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 (TiiS)* 8, 3 (2018), 1–25.

[13] Yefeng Liu, Darren Edge, and Koji Yatani. 2013. SidePoint: A Peripheral Knowledge Panel for Presentation Slide Authoring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, New York, NY, USA, 681–684. https://doi.org/10.1145/2470654.2470750

[14] B. J. Rhodes and P. 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

[15] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2010. Potential for Personalization. *ACM Trans. Comput.-Hum. Interact.* 17, 1, Article 4 (April 2010), 31 pages. https://doi.org/10.1145/1721831.1721835

[16] Tung Vuong, Salvatore Andolina, Giulio Jacucci, and Tuukka Ruotsalo. 2021. Does More Context Help? Effects of Context Window and Application Source on Retrieval Performance. *ACM Trans. Inf. Syst.* 40, 2, Article 39 (sep 2021), 40 pages. https://doi.org/10.1145/3474055

[17] Tung Vuong, Salvatore Andolina, Giulio Jacucci, and Tuukka Ruotsalo. 2021. Spoken Conversational Context Improves Query Auto-Completion in Web Search. *ACM Trans. Inf. Syst.* 39, 3, Article 31 (may 2021), 32 pages. https://doi.org/10.1145/3447875

[5] Partially funded by the EU H2020 project CO-ADAPT, the MIUR (PON AIM), and the Academy of Finland (322653, 328875, 336085, 319264, 292334).