A Retrospective Context Mining Approach For Bootstrapping Personal Knowledge Assistants

Desiree Heim[1,2], Christian Jilek[1,2], Heiko Maus[1] and Andreas Dengel[1,2]

[1] *Smart Data and Knowledge Services Department, German Research Center for Artificial Intelligence (DFKI), Trippstadter Straße 122, 67663 Kaiserslautern, Germany*

[2] *Department of Computer Science, TU Kaiserslautern, Erwin-Schrödinger-Straße 52, 67663 Kaiserslautern, Germany*

Abstract

Smart assistants supporting knowledge workers in their daily work are in demand nowadays. To provide individually tailored support, the assistants first have to gain knowledge about the knowledge worker and their information space. In our setting, the assistant should support the user according to their current mental context. To build the required mental context base, other approaches observe the human's interaction with information items like files, emails, etc. and infer contexts from this activity data. However, those procedures suffer from a cold start problem as the context base on which the assistant relies on is built alongside the observation. The context-mining approach introduced in this paper addresses this issue by relying only on document information that is available at the start-up time.

Keywords

Context Mining, Personal Information Management, Personal Knowledge Assistants

1. Introduction

Knowledge workers face an ever-increasing flood of information in their daily work. While AI-based personal knowledge assistants (PKA) are increasingly helpful in supporting users (e.g. cognitive load reduction, higher degrees of automation, etc.), they typically still suffer from the so-called cold start problem: When first started, the assistant does not "know" anything about the user. Thus, it takes a while of observing user activities until support measures by the system like recommendation, ranking, filtering, etc. become meaningful. Dragan and Decker [1], for example, assume that this was one of the problems why approaches like the Semantic Desktop [2, 3], although proven to be superior to traditional systems [4], were and are still not widespread.

A key concept for such assistants is user context: Especially in the aforementioned area of the Semantic Desktop, the Personal Information Model (PIMO) [5] is used to represent the user's mental model in a machine-understandable way. More recently, PIMO was extended to especially also represent users' various contexts as so-called context spaces [6]. We see context as a "sense-giving environment" for a (given) nucleus like an activity, an event or an

LWDA'22: Lernen, Wissen, Daten, Analysen. October 05–07, 2022, Hildesheim, Germany

desiree.heim@dfki.de (D. Heim); christian.jilek@dfki.de (C. Jilek); heiko.maus@dfki.de (H. Maus); andreas.dengel@dfki.de (A. Dengel)

0000-0003-4486-3046 (D. Heim); 0000-0002-5926-1673 (C. Jilek); 0000-0003-3508-5860 (H. Maus); 0000-0002-6100-8255 (A. Dengel)

information item itself (a document, email, etc.) [7]. Such contexts typically evolve over time, e.g. a large research task could spawn from a small context having only an email calling for participation as its nucleus [7]. Exploiting this contextual (meta-)information allowed for new ways of supporting users, for example by means of Managed Forgetting [8] which involves temporal hiding, reorganization, condensation, and similar measures.

This paper presents an approach for mining such user contexts to bootstrap a PKA. In contrast to all existing approaches found in literature, we solely rely on data available at a concrete point in time, the assistant's start-up time. Therefore we refer to our approach as a *retrospective* context mining approach as opposed to those tracking users from the start-up time on. Due to the retrospective nature, the approach can only exploit the information contained in the current static setting of the information space which is the result of numerous past interactions with the information items. The missing interaction information in the retrospective setting like window switches between documents, etc. can be seen as implicit hints from the user that certain documents may belong together [9]. Thus, in our approach, finding related documents requires a more exhaustive search of possibly related information items as those hints are not available. However, the exhaustive search has the benefit that we may find relations between documents that are not (easily) visible looking at the interaction history. Since our approach uses no implicit feedback through interaction, the knowledge worker is encouraged to provide feedback to improve the tool's suggestions. In the following, we also refer to our approach as the Contextifier.

The remainder of this paper is structured as follows: Section 2 gives an overview of related approaches and classifies our approach. Section 3 introduces our proposed retrospective context mining approach and Section 4 describes an early evaluation conducted to get first impressions of the approach's usability and perceived quality of the results. In Section 5, we discuss changes that are beneficial to incorporate into our approach before performing a larger study. The paper is concluded in Section 6 with a short summary and an outlook on the next steps.

2. Related Work

To effectively support knowledge workers during their daily work, it is manifest to first analyze their work setting and understand how their information space is structured. The extracted knowledge can then be used to tailor supporting mechanisms individually to the user, e.g., by recommending documents to them that could be helpful for their current context. In our approach, we aim to approximate the user's mental model of their work setting in the form of contexts which can then be further used as a basis for personal assistant systems supporting the user's knowledge work.

Most other approaches, that aspire to get a representation of information spaces, structure documents into tasks, activities, or topics. While the concepts of tasks and activities are similar to our context definition, topics do not necessarily translate into contexts as they are not user-centered but rely solely on textual features (see e.g., [10]). In the following, we speak of context structures or contexts explicitly including the closely related concepts of tasks, activities, workflows, etc.

The existing approaches identifying user-centered context structures can be categorized into three main categories. Manual approaches completely rely on the users who have to organize their documents into context structures added to the traditional folder structures [11, 12]. Semi-automatic and automatic approaches are less effort-intense than manual approaches, and all found (semi-)automatic approaches track the user's interaction with their documents to identify contexts. The main difference between semi-automatic and automatic approaches concerns the features used to divide the interaction event stream into contexts. Semi-automatic tools require the user to initially annotate contexts explicitly such that Machine Learning models can be trained based on the labeled data to be able to, later on, predict context labels automatically [13]. To enable a user feedback-independent context identification, automatic approaches often consult other features, like the documents' contents or their metadata like e.g., their title or folder location, to divide the interaction event stream into context structures [14, 15, 16, 17]. Some automatic approaches though only rely on interaction data and e.g., judge document similarity strengths based on how often the documents co-occurred [18, 19, 20].

Using manual approaches has the advantage, that one can be sure that the resulting contexts represent the user's mental model well as the user themselves created them. However, manual context creation is time-consuming and an overhead in addition to the actual knowledge work. Thus, automatic approaches are in most cases preferable. Nevertheless, in particular, semi-automated activity-based approaches but also automatic activity-based approaches suffer from the cold start problem meaning that it takes some time until they captured enough (labeled) interaction data to have a strong-enough basis of contexts to effectively support the user.

Unlike, to the best of our knowledge, all other existing automatic context mining approaches, our Contextifier does not rely on interaction event streams. Instead, our approach only takes the document's contents and all static metadata available at the tool's start-up time, like e.g., the last usage dates, the document's folder, or the previous email in the email's thread, into account.

As Context Mining is similar to Process Mining, it is crucial to highlight the differences between the two approaches in order to concretise the notion of Context Mining. Process Mining aims to model, monitor, and enhance real-world end-to-end processes by extracting knowledge from event streams [21]. However, both techniques differ with respect to their focus and goals. While Context Mining is centered around data and clusters documents according to their dedicated knowledge worker context, Process Mining is centered around processes and documents are considered only as input or output of activities. Similar to Process Mining, also Context Mining might take event streams as input and identify event sequences forming activities in the stream. However, their notion of activities is different. In Process Mining, activities are "well-defined steps in some particular process"[21] whereas activities in Context Mining are typically considered as individual entities which might be related to each other. Furthermore, our concept of contexts is more general as contexts do not need to be activities but can be also unstructured collections of documents that e.g. belong to a certain project.

3. Approach

3.1. Overview

Our context mining approach can be separated into four main phases (see Fig. **??**). In the input

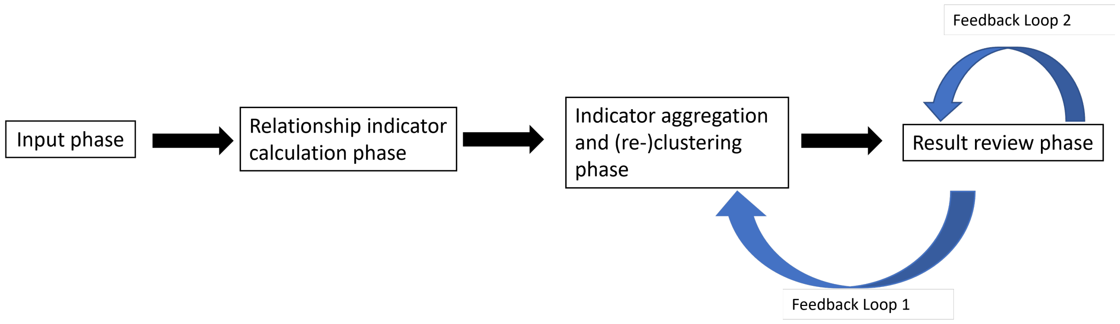
Feedback Loop 2
Input phase
Relationship indicator calculation phase
Indicator aggregation and (re-)clustering phase
Result review phase
Feedback Loop 1

Figure 1: Phases of the context mining process. Feedback opportunities are depicted in blue.

phase, the user provides their personal information items that should be considered or metadata of the items. Emails are accessed on the mail server via the *Internet Message Access Protocol (IMAP)*[1] and calendar entries are extracted from the provided *iCalendar*[2] file. Metadata of files and bookmarks are obtained by a crawling tool [22].

After the required information is given, the core context mining process starts with the Relationship Indicator Calculation phase. This phase aims to identify relationships between pairs of information items. For each document pair, multiple relationship indicators are calculated. The indicators are then aggregated such that a multigraph, composed of items as nodes and relationship indicators as weighted edges, becomes an ordinary graph only having at most one weighted edge between each node pair. The weighted graph, representing document relationships weighted according to their confidence, is then used to cluster the items. Finally, the clusters are named and interpreted as contexts.

In the Result Review phase, the user can explore the calculated relationships and contexts. Moreover, they can provide feedback to fine-tune the results. Enabling the user to be part of the calculation process is especially useful in retrospective context mining approaches because of the missing information about the user's interactions with the information items in the past. However, feedback is not mandatory, hence contexts can be identified completely unsupervised without any human intervention. Thus, providing feedback is only possible in the input phase and the last phase to ensure that the entire calculation process can run without any interruption.

3.2. Relationship Indicators

In the retrospective setting, several information item attributes are available. Besides the folder hierarchy and some interaction time information, like the last modification date or the sent date, the textual content and the titles of the items can be used as comparison attributes to examine the documents' relationships. Additionally, depending on the item type, also other information like involved persons, the answered email, the location of a calendar event, or attachments can be utilized.

[1] https://www.rfc-editor.org/rfc/rfc9051.html
[2] https://www.rfc-editor.org/rfc/rfc5545.html

Our department has a long history of research on information management (IM) and knowledge work. Based on this experience, we constructed several attributes that could serve as relationship indicators. Additionally, we used a questionnaire to also include some views and IM habits of external persons having different backgrounds. The eleven participants, students aged between 21 and 27 with an almost equal distribution between sexes, were asked to rate indicators according to their significance. The indicators were provided as statements about the relationship of document pairs e.g., "Two emails belong together if one email is a response to the other email". For each indicator, the participants could state whether they think it is helpful, partially helpful, or misleading. In addition, the participants could provide supplementary comments to further explain their ratings.

The results of this small supplementary questionnaire showed that the experience-based indicators we assembled as a starting point already covered external persons' feedback making them a suitable base of the current approach (later revision is possible though).

Besides the rating of the indicators, the comments suggest that it is important to incorporate multiple indicators in the relationship evaluation as regarding only a single indicator could be misleading in several situations.

Given that information spaces usually contain thousands of documents, it is crucial to compare them in a resource-preserving manner. To reach this goal, the indicators are chosen to be expressive but not too resource-intense. For each comparison aspect, we collected different measurement escalation levels. Measurements on a high escalation level contain more information but require more computational resources. Deciding for a measure on the next highest escalation level leads to higher computational costs. However, the additionally gained information content might be marginal such that choosing the lower escalation level is a reasonable trade-off.

An exemplary indicator to examine the relationship between a file and an email is the "is-attachment-of" indicator which states that documents are similar if file's and the attachment's titles are equal. The confidence score of this relation is given by the textual overlap of the file and the attachment. This overlap is calculated by one minus the Levenshtein distance [23] of the documents' contents divided by the maximum character count of the two texts. In this example, measurements on lower escalation levels are just checking the equality of the file's and the attachment's title or alternatively checking for documents with the same title whether their file contents are identical consulting by the files' sizes. However, just comparing the titles and additionally checking the content equality is quite restricting as attachments could be, for example, forms to be filled out such that the two documents' contents slightly differ. Furthermore, the downloaded attachment could have been renamed such that the previously introduced measurements can not identify the relationship. Without the precondition that the attachment and the file must have the same title, the algorithm would have to check the attachment's content against all files. As this is computationally expensive and might only identify a few more relationships, the initially introduced measurement is chosen.

Besides the attachment indicator, the folder hierarchy distance, the access time closeness, and the text, title and folder title similarity serve as indicators. Additionally, the email response relation, the occurrence of a bookmark URL in a document, the equivalence between calendar file email attachments and calendar entries, the involved persons in an email or calendar entry, and the location similarity between calendar entries are taken into account as more type-specific indicators. For each document pair, a subset of the mentioned indicators is calculated as not all indicators apply to all types of document pairs.

After having evaluated all document pairs, we obtain a multigraph, in which documents are the nodes and the different relationship indicators the edges weighted according to the indicators' confidence values.

3.3. Aggregation and Clustering Of Identified Relationships

During the aggregation phase, the resulting multigraph is turned into an ordinary graph with at most one edge between two nodes. To aggregate the already normalized indicator values, the Contextifier calculates a weighted sum whereas every value receives a weight between zero and one. These weights are set per document pair type and sum up to one to obtain a normalized aggregated relationship confidence. Besides the weights determining the relative importance of the indicators, "particularly important indicators" overrule the weighted sum if their value is higher than the one of the weighted sum. Thus the aggregated relationship confidence *rc* of a document pair *p* with the weights w_i, relationship indicator values v_i, the pair type-specific relationship indicator set R and the set of "particularly meaningful indicators" PMI can be described by the following equation:

$$rc(p) = max\left(\left\{\sum_{r \in R(type(p))} w_r * v_r(p)\right\} \cup \bigcup_{pm \in PMR(type(p))} v_{pm}(p)\right)\Bigg| \sum_{r \in R(type(p))} w_r = 1, v_i(p) \in [0,1]$$

After the calculation of the relationship graph, the Contextifier clusters the documents accordingly. Initially, all documents are put into distinct clusters. According to the hierarchical agglomerative single-linkage clustering algorithm, the clusters of the two documents with the minimum relationship distance are merged in each iteration step as described by Jain et al. [24].

The single-linkage is especially well-suited as it is easier to find the next clusters to be merged compared to complete-linkage clustering or average-linkage clustering. Using single-linkage clustering, it is possible to sort a list of relationships according to their strength, search for the clusters to which the two documents with the strongest relationship belong, and merge them. Complete-linkage clustering merges clusters according to the minimal maximum distance between two documents of two distinct clusters. Thus, it requires looking at the relationship between all pairs of documents of distinct clusters and does not allow just looping over the sorted relationship list. Average-linkage clustering merges clusters according to the minimum average distance between all documents of two clusters and thus also requires the consideration of all document relationships between clusters.

Single-linkage and complete-linkage clustering have the weakness that one single relationship might have a high impact on the resulting clusters. In the case of complete linkage, one weak relationship might prevent two otherwise similar clusters to be merged. Single-linkage clustering suffers from so-called "bridge effects" [24] which means that two clusters might be not very similar but are merged because of a single strong relationship between two documents. As in our scenario, the relationship value is composed of multiple indicator values, the risk of bridge effects is reduced due to the relatively low chance that all or most indicators imply a high relationship for unrelated documents.

In summary, single-linkage clustering is suited to our scenario as it is particularly resource-preserving. The obtained clusters are in the last calculation step interpreted as contexts and named accordingly.

3.4. Interaction Possibilities

After the calculation, the user can inspect the clustering results in two separate views. The first view, the Tree View, is structured as a file explorer so that the user can choose a context on the left panel and gets details about the contained documents. In the alternative view, the user can explore the result landscape via a network depicting documents as nodes and the accumulated relationship between them as edges. The nodes in the network are placed according to their edge strength and therefore allow to inspect the document similarities visually. Moreover, nodes are colored according to the corresponding context of the represented document. Details about the documents and their context are visible after clicking on the respective node. Both views offer in principal the same functionality except the used visualization methods differ.

In addition, the user can see the calculated relationships of each document to others and inspect a specific relationship further to get more insight into how the relationship confidence was calculated. When inspecting the relationship, the applied relationship indicators, their values, an additional explanation of the indicator values, the weights of the indicator, and the information on whether it is marked as a "particularly meaningful indicator" are shown.

3.5. User Feedback

As the retrospective nature of this approach does not have access to any direct or indirect user feedback, as it is the case for active approaches tracking the user's interactions, a result refinement by the user is beneficial.

During the result inspection, the user can either directly or indirectly manipulate the resulting contexts. If the user has the impression that the initial set-up of relationship indicator weights or the "particularly important indicators" did not work well on their data, the user can change those settings and initiate the indicator aggregation and clustering process again. Moreover, contexts can be merged, split, deleted, or renamed and documents can be moved into another context or deleted completely from the context abstraction layer.

4. Early Evaluation

In an early evaluation we conducted, the usability of the Contextifier, the quality of the identified relationships and contexts, and the exploration possibilities using the web interface are accessed. Fourteen participants took part in the study. Six of them are members of our working group at the DFKI and the remaining ones are students. Five of the DFKI employees that participated were male and one female and were between mid-twenty and early fifty years old. Most of the three female and five male participating students had a Computer Science background and were between 21 and 27 years old. The participants could express their feedback through rating on Likert scales and additional supplementary free text comments which allows to get more insight

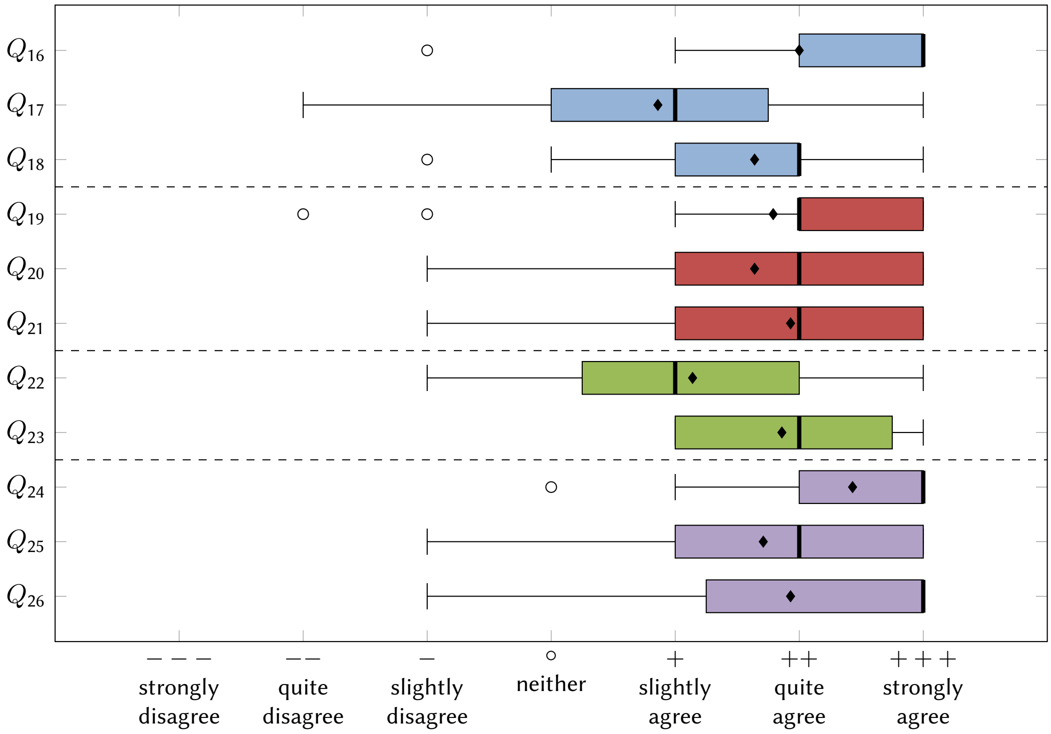

Figure 2: Survey Result for Custom Items

into the participants' reasoning behind their ratings and thus come up with more informed interpretations of the results.

To get a first impression of the tool's usability, we used the standardized Computer System Usability Questionnaire (CSUQ) [25]. The questions of this questionnaire assess the system's usefulness, information quality, and overall usability on a 7-point Likert scale where high values reflect a high agreement with a statement. Questions 3, 4, 5, and 14 of the CSUQ questionnaire are excluded in our evaluation as they target the usability and value of a system in business processes which does not match our application scenario as the focus of the Contextifier is the extraction of contexts and not the further usage of those mined contexts. Hereby, the order of the incorporated questions was preserved.

Overall, the usability was rated mainly positively. However, the results indicate that the error handling and error recovery have to be improved.

Additional to the usability, the quality of the initial results, the transparency of the calculation process, the ease of finding the right information, and the interaction possibilities are assessed in the study. The questions targeting those criteria are specific to our tool. To keep the answer format consistent, the participants also had to indicate their level of agreement on the 7-point Likert scale. The detailed ratings of the tool-specific questions are depicted here in a box plot (see Figure 2). The diamond symbols represent the means and the perpendicular lines the median.

The quality of the initial results, i.e. the results obtained before any user feedback, were rated with respect to the perceived intra-context similarity (Q16), the context labels (Q17), and the relationship confidences (Q18).

On average the questions Q16-Q18 received a good score but the results show that improvements are possible. This is not surprising as we only have limited indirect feedback, mainly in form of the different folder hierarchies, from the user during the first calculation due to the retrospective nature of our approach and the consequently limited availability of data about the user's information item interaction. To cope with that, our tool allows the user to further refine the results. Moreover, context labels could be improved.

Our tool should not only identify relationships between documents and cluster them into contexts but it should also explain the results to the user. Relationships between documents are explained by showing the user how the specific relationship confidence was calculated by showing the applied relationship indicators, their values, and their weights. The perceived transparency was evaluated regarding the comprehensibility of the documents' relationship confidences (Q19), the context composition (Q20), and the effect of the aggregation parameters, namely the relationship indicators' weights, and annotations as "properly important" (Q21).

The overall result indicates that the users are satisfied with the transparency.

The third question category specific to our tool is the ease of finding the right information. It was assessed by ratings of the general ease of finding the desired information (Q22) and the information content of the different result views (Q23).

The participants rated the ease of finding the right information rather positively.

As user feedback is important in our approach to refine the results as we only have limited details about the user's interaction, we want to enable the user to give meaningful feedback. The interaction possibilities could be judged by rating the available direct feedback operations, like renaming, merging, etc. (Q24), the indirect feedback operations, like the adaption of relationship aggregation parameters (Q25), and the completeness of the offered feedback operations (Q26).

The good ratings for the interaction possibilities indicate that the feedback options available in the first version were already working well and the expected core operations were included.

As the Contextifier user interface offers two different views to inspect the results and provide feedback, the advantages, and disadvantages of the views are also examined in the study.

The results show no clear preference of the participants for one of the two views and the detailed results were therefore omitted here. The Tree View is perceived as easier to use and less overwhelming. Whereas, the Graph View delivers more information by incorporating the relationships between documents in the network visualization, depicts the results more comprehensibly, and provides a better overview of the result landscape. Both alternatives have their strengths and complement each other well.

In summary, the results of this early study indicate that the Contextifier delivers meaningful contexts, which can be well refined using the available feedback operations. Nevertheless, there is still potential for improvement, especially regarding usability aspects which should be addressed before conducting a larger study to avoid that usability problems have a negative impact on the tool's main functionality. In the next section, we discuss several aspects that should be improved and how these first evaluation results can be interpreted.

5. Discussion

The early evaluation of our approach and especially the provided complementary comments of the participants show several improvement potentials. In particular, the performance of the calculation process and the scalability of the Graph View have to be raised such that users can input a larger document set. To improve the calculation performance, the rule calculation process should be extended by a preceding step. In this step, the documents should be first assigned to groups containing similar elements. The initial group assignment can be geared to folders and groups can be split until the intra-group similarity is acceptable. Groups can then be compared to each other analogously to the indicator-level rule calculation. If the Contextifier identified a potential relationship between two groups, their respective information items should be inspected in more detail by calculating relationship indicators on the item level between pairs of information items belonging to respectively one of the two groups. This additional step would limit the number of item-level comparisons and thus the resource consumption of the calculation process while still identifying the most strong document relationships.

To improve the scalability of the Graph View, the network's abstraction level can be changed from the information item level to the context level, i.e. depicting contexts and their relationships instead of documents and their connections. This would lead to a much faster network generation and would additionally reduce the complexity of the depicted information such that users get less overwhelmed.

Regarding the transparency of the calculation process, some users requested more information about the contexts, e.g. how they developed over the clustering iterations. Additionally, the inter-context similarity could be shown which is integrated in the revisited Graph view suggested above.

Moreover, participants requested some extensions of the current feedback mechanisms. Merging and splitting could be possible for more than two initial or respectively resulting contexts at a time. Some participants also expressed their willingness to provide pre-feedback before the calculation process to exclude folders containing messy or unimportant information and to give hints on which folders might be good context candidates. Additionally, it was requested to have the possibility to add documents, belonging to the same folders as documents in a context, to that context manually. This would allow to also include documents that were, according to the relationship indicators, not identified as similar to a context's documents and thus make use of the user's expert knowledge to get a higher amount of documents assigned to contexts.

Despite several aspects that should be improved before conducting a more elaborated user study, the first impressions gained by the study indicate that the approach is promising and it is possible to extract meaningful contexts in a purely retrospective setting. Thus, these contexts can serve as a basis for personal knowledge assistants to provide tailored support to users. Moreover, (semi-) automatic activity-based approaches, such as those presented in the Related Work Section, might also profit by incorporating the obtained contexts as an initial context basis such that they have prior knowledge about the user's context landscape and can refine it using activity information.

6. Conclusion and Outlook

In this paper, we introduced an approach to mine knowledge worker contexts. In contrast to other approaches, we only make use of static information, the traces of the user's past interaction with their information items. This retrospective orientation allows to bootstrap contexts and thus avoids a potentially long cold start phase of a personal knowledge assistant. Despite the sparsity of the available information, the participants in our early experiments indicate that the resulting contexts are meaningful and could be refined well using the available feedback operations. Based on the insights gained, we also discussed several improvements that should be implemented before conducting a larger study. An interesting extension of our approach would be the introduction of context of different depths to get a context hierarchy instead of a flat set of contexts. Furthermore, it could be examined whether a prior automatic selection of promising folders as initial context candidates before starting the detailed calculation process is beneficial. Moreover, future experiments could access the benefit of incorporating contexts extracted by our tool as prior knowledge into activity-based (semi-) automatic Context Mining approaches.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research in the project SensAI (grant no. 01IW20007). The authors would like to thank Jessica Chwalek and the rest of the SensAI team as well as the participants of the user study for their contributions.

References

[1] L. Dragan, S. Decker, Knowledge management on the desktop, in: Knowledge Engineering and Knowledge Management, Springer, 2012, pp. 373–382.

[2] S. Decker, M. Frank, The Social Semantic Desktop, TR 2004-05-02, DERI, 2004.

[3] L. Sauermann, A. Bernardi, A. Dengel, Overview and outlook on the semantic desktop, in: Proc. ISWC 2005 WS on The Semantic Desktop, volume 175, CEUR-WS, 2005, pp. 74–91.

[4] T. Franz, A. Scherp, S. Staab, Are semantic desktops better?: Summative evaluation comparing a semantic against a conventional desktop, in: Proc. 5th Int'l Conf. on Knowledge Capture, K-CAP '09, ACM, 2009, pp. 1–8.

[5] L. Sauermann, L. van Elst, A. Dengel, PIMO – a framework for representing personal information models, in: Proc. of I-Semantics '07, Know-Center, Austria, 2007, pp. 270–277.

[6] C. Jilek, M. Schröder, S. Schwarz, H. Maus, A. Dengel, Context spaces as the cornerstone of a near-transparent and self-reorganizing semantic desktop, in: The Semantic Web: ESWC 2018 Satellite Events, Revised Selected Papers, Springer, 2018, pp. 89–94.

[7] P. Gauselmann, Y. Runge, C. Jilek, C. Frings, H. Maus, T. Tempel, A relief from mental overload in a digitalized world: How context-sensitive user interfaces can enhance cognitive performance, Int'l Journal of Human-Computer Interaction (2022). In press.

[8] C. Jilek, Y. Runge, C. Niederée, H. Maus, T. Tempel, A. Dengel, C. Frings, Managed

forgetting to support information management and knowledge work, KI – Künstliche Intelligenz 33 (2019) 45–55.

[9] P.-A. Chirita, S. Costache, J. Gaugaz, W. Nejdl, Desktop context detection using implicit feedback, in: Proc. SIGIR 2006 WS on Personal Information Management, 2006, pp. 24–27.

[10] R.-L. Liu, Y.-L. Lu, Incremental context mining for adaptive document classification, in: Proc. 8th ACM int'l conf. on Knowledge discovery and data mining, 2002, pp. 599–604.

[11] V. Kaptelinin, Umea: translating interaction histories into project contexts, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 2003, pp. 353–360.

[12] G. Smith, P. Baudisch, G. Robertson, M. Czerwinski, B. Meyers, D. Robbins, D. Andrews, Groupbar: The taskbar evolved, in: Proceedings of OZCHI, volume 3, 2003.

[13] S. Stumpf, X. Bao, A. Dragunov, T. G. Dietterich, J. Herlocker, K. Johnsrude, L. Li, J. Shen, Predicting user tasks: I know what you're doing, in: 20th National conf. on artificial intelligence (AAAI-05), WS on human comprehensible machine learning, 2005.

[14] N. Oliver, G. Smith, C. Thakkar, A. C. Surendran, Swish: semantic analysis of window titles and switching history, in: Proc. 11th int'l conf. on Intelligent user interfaces, 2006, pp. 194–201.

[15] R. Lokaiczyk, A. Faatz, A. Beckhaus, M. Goertz, Enhancing just-in-time e-learning through machine learning on desktop context sensors, in: International and interdisciplinary conference on modeling and using context, Springer, 2007, pp. 330–341.

[16] J. Makolm, S. Weiß, D. Reisinger, Dyonipos: proactive knowledge management, BLED 2008 proceedings (2008) 10.

[17] B. Schmidt, J. Kastl, T. Stoitsev, M. Mühlhäuser, Hierarchical task instance mining in interaction histories, in: Proc. 29th ACM int'l conf. on Design of communication, 2011, pp. 99–106.

[18] C. Abela, C. Staff, S. Handschuh, Automatic task-cluster generation based on document switching and revisitation., in: UMAP Workshops, 2015.

[19] O. Brdiczka, From documents to tasks: deriving user tasks from document usage patterns, in: Proc. of the 15th int'l conf. on Intelligent user interfaces, 2010, pp. 285–288.

[20] K. Gyllstrom, C. Soules, A. Veitch, Activity put in context: identifying implicit task context within the user's document interaction, in: Proc. 2nd Int'l Symposium on Information Interaction in Context, 2008, pp. 51–56.

[21] W. v. d. Aalst, A. Adriansyah, A. K. A. d. Medeiros, F. Arcieri, T. Baier, T. Blickle, J. Bose, P. v. d. Brand, R. Brandtjen, J. Buijs, et al., Process mining manifesto, in: Int'l conf. on business process management, Springer, 2011, pp. 169–194.

[22] M. Schröder, C. Jilek, A. Dengel, Interactive concept mining on personal data - bootstrapping semantic services, CoRR abs/1903.05872 (2019).

[23] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet Physics Doklady, volume 10, 1966, pp. 707–710.

[24] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM computing surveys (CSUR) 31 (1999) 264–323.

[25] J. R. Lewis, IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use, Int'l Journal of Human-Computer Interaction 7 (1993) 57–78.