Effective Data Mining Support for Personal Information Management

Korinna Bade
Anhalt University of Applied Sciences, Dep. of Computer Science and Languages
Lohmannstraße 23, D-06366 Köthen (Anhalt), Germany
k.bade@inf.hs-anhalt.de

Marcus Nitsche
Otto-von-Guericke-University Magdeburg, Department of Computer Science
Universitätsplatz 2, D-39106 Magdeburg, Germany
marcus.nitsche@ovgu.de

Andreas Nürnberger
Otto-von-Guericke-University Magdeburg, Department of Computer Science
Universitätsplatz 2, D-39106 Magdeburg, Germany
andreas.nuernberger@ovgu.de

ABSTRACT

Information management is a tedious task that we wish to support through automatic methods. In the past, we developed different data mining techniques capable of providing automatic structuring to be used in information management. In this paper, we discuss how those algorithms can be employed in a system assisting the user through appropriate visualization and interaction. We identify requirements on the system, propose several design elements and describe how they enable effective interaction between the user and the data mining methods. Those form the base for future user studies.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

Keywords

personalization, constrained clustering, hierarchical structuring, user interface design, information management

1. INTRODUCTION

Managing information and keeping them for later use is often a tedious task, which users try to avoid as much as they can afford [7]. Nevertheless, a good organization scheme is often a prerequisite for re-finding information later on, especially if the size of the information collection increases. Therefore, our goal is to provide the user an effective and efficient tool for information organization, embedded in his information retrieval and/or information usage process. Our focus lies in the development of appropriate data mining algorithms for this task and their effective integration in a personal information management tool. Due to the fact that data mining algorithms are often very complex, it is crucial to provide a user interface, which hides this complexity, however, at the same time communicating the effects on his data towards the user and offering him the necessary means to take corrective action and control those algorithms.

A simple example, in which data mining is efficiently employed in information management, is automatic junk mail classification in email applications. Nowadays, this is standard functionality accepted by users although errors are made by the classifier. One reason for this is that the user is still in control over his data and can take corrective action. Our goal is to carry this methodology over to the more complex task of information structuring.

Currently, we are working on the support of hierarchically organizing a user's information. The reason for this is that we regard hierarchies as a very powerful tool for organizing one's information space, if data load is high. And with the steadily increasing amount of digital data available and further produced every day, this will be the common case. The biggest draw-back of hierarchies is their continuous management [7]. Therefore, we aim at simplifying this management through (semi-)automatic tools. The employed background algorithms doing the work are discussed in Section 2.

We envision an integrated tool for (personal) information organization on the one hand and information retrieval and/or usage on the other hand. As an example, consider the browsing application sketched in Fig. 1. On the top, you find the navigational part where you can either enter the URL of a website or issue a search query. The rest of the design is split in two sections. On the left, the user's information space is visualized. In this example, it is a bookmark hierarchy visualized through an ordinary folder tree. However, this could be replaced by more sophisticated visualization techniques like a mindmap visualization [6]. The right side is used for the user's information usage displaying either the content of a selected website or search results. Please note that we use the browser application as an example throughout the paper. However, the same ideas are applicable to any kind of data like managing the files in the file system or emails or a merged view over all of these information pieces. In Section 3, we discuss how we envision visualization and interaction in the system and its combination with the underlying algorithms.

2. PERSONALIZED HIERARCHICAL STRUCTURING

Figure 1: Concept of an integrated browser

To provide automatic support for hierarchical organization of information pieces like documents, data mining techniques are required that can structure elements in a hierarchy. These can be hierarchical classification (supervised) or clustering (unsupervised) methods. However, standard methods of both types have different drawbacks in our scenario. Classification methods assume that elements to be classified belong to one specific node of a predefined hierarchy and stick to "inserting" elements in a node of that hierarchy. However, in our scenario, new documents might belong to a different topic not created by the user yet as he has never filed anything about it before. Clustering methods on the other hand have this flexibility as they are designed to generate structure where no structure was defined before. However, they ignore existing hierarchical structures which might lead to a totally different organization scheme than the one intended by the user.

Therefore, we developed and analyzed in our work different "in-between" algorithms which are semi-supervised. First, we developed different methods for hierarchical classification that take into account unknown topics and user interaction with the classification result [2, 3]. Second but more appropriate for our scenario, we developed constrained, hierarchical clustering methods that can hierarchically structure a collection of documents while taking into account an existing hierarchical structure [4, 5]. Hence, the result of the clustering is an extended user hierarchy. The original structure can be recovered by the user. However, new nodes or entire sub-hierarchies are added to present an overall organization scheme that is also appropriate for the new data. This enables the user to view the new data in relation to his existing information space, which is an organization scheme he is familiar with. This idea is visualized in Fig. 2.

Coming back to our integrated browser example, the given hierarchy on the left of Fig. 2 is the current information space of the user on the left of Fig. 1. The document collection on the right of Fig. 2 could represent the result set of a web search but also a single document currently viewed by the user. Our developed algorithms then propose a means to organize this new data on the right into the existing information space on the left using existing nodes where possible and generating new nodes (A, B, and C) when necessary.

We developed and evaluated our algorithms under the specific requirements given by our scenario in which users interact with their own and external data. The results are very encouraging for further research. [1] provides a complete view on the whole topic of personalized hierarchical structuring and all results in comparison.

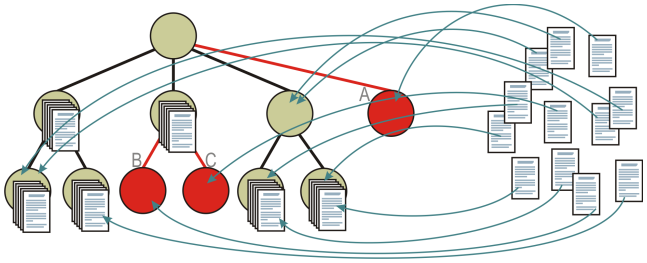

Figure 2: Constrained hierarchical clustering

3. VISUALIZATION AND INTERACTION

However, the best data mining algorithms are useless, if they cannot be employed effectively by the user. Therefore, we need to develop an appropriate tool that usefully integrates our methods in assisting information organization and retrieval. In the following, we discuss the requirements for visualizing the information space and its modifications through the algorithms as well as for user interaction. Furthermore, we present some ideas about useful interaction and design elements that could be employed.

3.1 Requirements

There are several requirements on a system with automatic structuring support. First of all, the goal of the system is to simplify the management of information. Hence, the system should not increase the user's burden by requiring more clicks while performing his usual tasks. Therefore, it is desirable if the system uses implicit user feedback according to performed structuring suggestions as much as possible. At the same time, the user must always be in control over his data. Automatic structuring through the system should be handled as a suggestion as long as the user does not indicate that he fully agrees with it (either explicitly or through strong implicit evidence). Any change performed through the system must be clearly communicated to the user and needs to be reversible through user action. This ensures traceability.

There are two main tasks that the system should offer for automatic user support. First, it must be able to automatically structure new documents according to the existing user information space to assist filing of new information. Second, it must support a continuous management of the information space to ensure an up-to-date hierarchy. For the first task, the automatic method modifies the existing information space to add new documents possibly creating new folders as well. This is a suggestion that might be temporary or faulty and should be visualized as such. Desirable is an integrated view on the user's information space and the suggested modification, which we call the *extended information space*. Agreeing to the suggestion (i.e., allowing the system to add information to the user's information space) should be simple for the user, not be required immediately, and not necessarily explicit but also through usage. Disagreeing should not create extra burden for the user which he would not have had without the system.

The first task usually "just" extends the current user's information space. In contrast to this, the second task involves re-structuring like moving misplaced elements or deleting folders. This becomes necessary, if documents were filed inconsistently or the user's filing behavior changed over time. However, re-structuring has a much greater impact on the user's perception and might lead to disorientation. Therefore, it should not be done too often and never without the user's approval or even only on explicit request. If re-structuring takes place, the (suggested) changes must be visualized, allowing the user to take corrective action, if necessary. Hence, the visualization must compare the original user information space with the modified one and allow user feedback about changes. With this feedback an improved information space is built in a semi-automatic way that is accepted by the user.

3.2 Design Ideas

In the following, we discuss several ideas on the user interface design that address the previously identified requirements. For the task of adding new elements to the information space through automatic methods, Fig. 3 shows different visualization techniques. Here, we present our idea of an integrated search that visualizes search results in strong relation to the user's information space. Search results are added to the information space as determined by the methods described in Section 2 and visualized as gray elements in italics following the principle of dual information coding [8]. This holds for documents as well as for folders created by the algorithms (e.g., the *Psychology* folder in Fig. 3). If added documents are hidden through closed folders in the visualized hierarchy, they are symbolized through counts given for those closed folders. Through this, the user always has an overview over the relations between search results and his information space and can directly navigate to interesting (new) documents. At the same time, the search results are displayed on the right as the usual ranked list. However, result information is augmented with a presentation of the hierarchical path in the information space determined through the automatic methods. This again provides a link to the user's information space. When inspecting the results, the user can choose between navigation on the left or right without loosing the connection between new and old information pieces. A further link is generated through a hover effect (as visualized for *Page N* in Fig. 3) that highlights the element in both visualizations and therefore builds a visual bridge between both sides.

If a user wants to add one of those results permanently to his information space, he can do so by clicking the apply hook of the element either in the information space or in the result list. In the latter, the user can choose to add the element at any level of the hierarchy through the respective hook. If the automatic classification was incorrect but the user wants to add this element to his information space anyway, he can do so by simply dragging it into the correct folder. Once added to the information space, the visualization of the element changes. On the left, it is shown in black. On the right, the path visualization is also given in black. Furthermore all hooks are removed. An example is page D in Fig. 3.

This modification of the information space is done based on explicit user feedback. Furthermore, we want to support modification through implicit feedback. This should

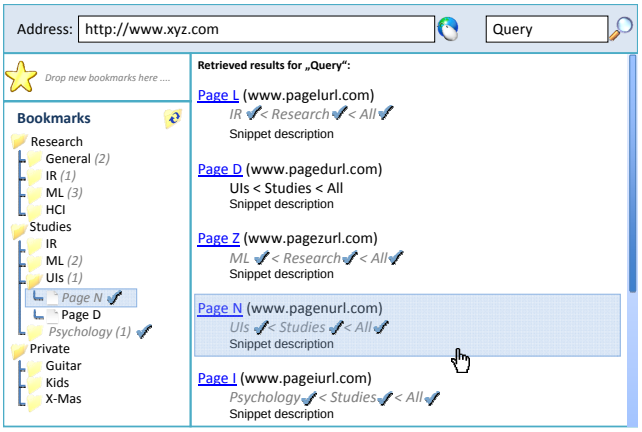

Figure 3: Concept of an integrated search

work as follows: If a user shows some interest in a document, e.g., by following a link and displaying the content of the page, it is added to his *extended information space*. This means the document will not be forgotten immediately through the system, even though the user has not explicitly stored it. Visualization changes by removing the italics but keeping the gray color and hooks. As the element continues to be displayed to the user in later sessions, he can further interact with that item. If he does not, he probably had no permanent interest in it. Through some aging mechanism, the element can be removed from the extended information space. However, if he shows further interest through interaction, the system is triggered to add the element to the permanent information space of the user. As this invades the user's data management, it has to be communicated to him. This can be done through two alternative methods (Fig. 4). First, the system could open a dialog asking the user, whether the item should be added. Second, the system could just display a toast notification and performs the change immediately (although reverting the change could be ensured through some other mechanism). While the first method means more direct user control over his data, the second method reduces the user's work load (in case that adding was correct) by requiring one click less.

Visualizing the external data through different text styles (gray, italics) is just one option. A different possibility is the use of different icons. In Fig. 5, we show an excerpt from Fig. 3 to demonstrate the use of icons. As before, we distinguish between temporarily added items from the search result list (on the top of Fig. 5) and items of the extended information space with which the user has started interacting (on the bottom of Fig. 5). Of course, it is also possible to combine both, text styles and iconic cues, especially if the user needs a strong visual differentiation between the different kinds of information pieces displayed. What works

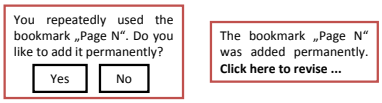

Figure 4: Communicating changes through a dialog (left) or a toast notification (right)

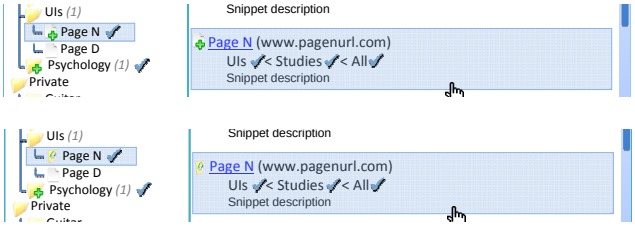

Figure 5: Different iconic cues (top: page N as temporarily displayed search result; bottom: page N as part of the extended information space)

best should be analyzed through a user study.

Adding every element to the extended information space with which the user has interacted, might blow up the size of the information space despite the use of aging techniques for removing elements which are not of interest. To avoid this and ensure only adding elements which are really of interest to the user, we propose the use of a "drop zone" as an alternative. You find the drop zone visualized in Fig. 1 and 3 above the bookmark hierarchy. The user can drag over elements to this region which he wants to store in his (extended) information space. If the element is not yet categorized in the information space, this is done now and the element is visualized as belonging to the extended information space. Further interaction works as described before. That means the user can permanently store the element through clicking the apply hook, change the location through drag-and-drop, or the system uses implicit feedback to automatically assign the element to the permanent information space of the user.

In the rest of this section, we discuss briefly a support of continuous management of the user's information space. Although the previously described adding techniques are supposed to reduce the problem of inconsistent filing of information pieces, the system should offer some means to clean such fragments. They might occur, e.g., through a change in filing behavior due to a new conception of the data by the user. This cleaning can be requested by the user through clicking the restructure button that you find in the top right corner of the bookmark hierarchy visualization. From the algorithmic side, old elements are reassigned to the information space by our data mining algorithms. If the computed position differs from the old one, the system suggests a change in the information space. All found changes are then visualized. This means not only giving the user the final result of restructuring but also showing what changed and how. Doing this in a way that is easy to understand by the user is a rather difficult task. Although several visualizations for hierarchies as well as differences between hierarchies exist [9], they are usually very difficult and rather suitable for an expert user than an ordinary user managing his files. It will be the focus of further research by us to develop an appropriate visualization technique.

4. CONCLUSION AND FUTURE WORK

With this paper we argue in favor of the use of appropriate data mining algorithms to support a user in information management. While the development of those algorithms is a challenging task in itself, it is not sufficient for effective user support. Its use in a system and the visualization of its results is very important. We showed different ideas on how such a visualization and interaction with it could look like. This is meant as a base for further discussion and for the design of respective user studies, which we plan to conduct in the near future. While visualizing extensions to an (hierarchical) information space is possible through easily understandable means, showing modifications that alter the underlying structure (including adding, moving and deleting of items) is a lot more challenging. We like to address this in detail in future work.

Furthermore, it is more and more important to design user interfaces in special consideration of the targeted device. While the proposed ideas may work fine in a desktop PC environment, they might not be suited for mobile devices like tablet PCs. As the usage of these devices is rapidly increasing, we also plan to develop the proposed system for tablets and analyze the suitability of different visualization and interaction methods under their requirements. Such a portable integrated tool for information organization and retrieval would enable a user to carry his personal information space with him and enables mobile personal interaction with information.

5. ACKNOWLEDGMENTS

The work presented here was partly supported by the German Ministry of Education and Science (BMBF) within the ViERforES II project, contract no. 01IM10002B.

6. REFERENCES

[1] K. Bade. *Personalized Hierarchical Structuring*. Sierke Verlag, 2009.

[2] K. Bade, E. Hüllermeier, and A. Nürnberger. Hierarchical classification by expected utility maximization. In *Proceedings of the 2006 IEEE International Conference on Data Mining*, 2006.

[3] K. Bade and A. Nürnberger. Rearranging classified items in hierarchies using categorization uncertainty. In *Proceedings of 30th Annual Conference of the German Classification Society*, 2006.

[4] K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 13–24, 2008.

[5] K. Bade and A. Nürnberger. Learning a metric during hierarchical clustering based on constraints. In *Proceedings of the LWA 2009 Workshop*, 2009.

[6] J. Beel, B. Gipp, and J. O. Stiller. Information retrieval on mind maps Ű what could it be good for? In *Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'09)*, pages 1–4, 2009.

[7] W. Jones. *Keeping Found Things Found - The Study and Practice of Personal Information Management*. Morgan Kaufmann, 2008.

[8] A. Paivio. *Mental representations: a dual coding approach*. Oxford University Press, 1986.

[9] H.-J. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, 2011.