Personalized Structuring of Retrieved Items

Korinna Bade
Faculty of Computer Science
Otto-von-Guericke-University Magdeburg
39106 Magdeburg, Germany
kbade@iws.cs.uni-magdeburg.de

Andreas Nürnberger
Faculty of Computer Science
Otto-von-Guericke-University Magdeburg
39106 Magdeburg, Germany
nuernb@iws.cs.uni-magdeburg.de

ABSTRACT

People nowadays struggle with huge unstructured collections containing some important knowledge. Search engines were developed to allow an easy access to relevant information, however they themselves produce for many queries large (unstructured) result sets. We suggest to tackle this problem by structuring these result lists or even complete collections in a personalized way, i.e. as the user would have structured it. To achieve this goal, we propose to use information about structuring behavior available from, e.g, an existing bookmark or folder hierarchy of the user. Besides structuring, the final visualization should also dependent on the user's preferences. Furthermore, we work on performance measures, which take the effect of the personalization into account to reduce the need for user studies.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

personalization, semi-supervised classification/clustering, performance measures, user profiling/grouping

1. INTRODUCTION

Storing data and making it available to a larger community (e.g. friends, business partners, or everybody in this world) has become very easy and cheap. This has led to a situation, in which everyone is confronted with an enormous amount of data, out of which he has to select only the pieces of information that are of value for him. In order to manage the data and with it the information contained in it, people usually try to organize or structure data to allow for a faster (re-)access. However, the degree, to which one can influence the structure of used data, varies.

One usually has total control over how personal files, emails, etc. are stored, although one depends on the capabilities of the used software, which might hinder that emails are stored together with text documents, images, etc. Shared company data or a shared folder of a work group is usually maintained by different people with different views of how the data should be organized. The influence of a single group member is therefore limited. The organization of other collections like the Web is (almost) uncontrolled and based on a self-organization process. Here, one has the possibility to manage the little seen part e.g. by creating bookmarks but the collection itself cannot be influenced. In addition, even there, where people can manage the data, they tend to loose track, e.g. because they postpone the organization and then forget about it or because they change their organizational structure in between.

This problem is nowadays tackled with search engines, which index today large parts of the Web and in the case of Desktop search engines personal files, emails, etc. The user expresses his information need with some keywords and gets a result list with possibly relevant documents. The problem that remains is that the result list itself tends to get very long for many queries issued today as the collection itself is so enormous. Furthermore, relevance is determined on general criteria like popularity of a page. However, what is relevant to a certain user depends on many different factors, e.g. his previous knowledge about the topic, the context, in which he issued the query, his cultural background, and a lot more.

In our opinion, a possible solution could be found in the personalization of the search. More specific, our idea is to structure the search results as the user would have done it. The structuring should basically have two goals. First, it should make it easy for the user to distinguish between relevant and non-relevant items on a large scope. This means the user can exclude a large set of items very quickly without the need to actually look at them. Second, the structuring should provide information about covered subtopics interesting to the user.

Besides the development of such algorithms, it is very important to be able to evaluate their quality. In the field of personalization, the most common way are user studies. However, they require a lot of effort and time to be prepared, executed, and evaluated. Furthermore, personalization might require a long-term interaction of the user with the system, which cannot be realized in a user study. Therefore, we are interested in developing performance measures and test scenarios that can capture aspects of the personalization, if possible, without requiring a user study.

2. USER PROFILING

A prerequisite for personalization is a user profile. In our case, it should ideally provide a complete image of the user's interaction with the data collection. This includes all items accessed and/or filed and when, the organizational structure created by the user for re-accessing the data, e.g. a bookmark hierarchy for web sites, and notes associated to single items or groups of items. Furthermore, it could include all search queries issued together with the final result of the search, i.e. accessed and/or filed result elements and their follow-ups, e.g. the important result of a web search could be found not directly on a result page but after following a link from a result page to another page.

From this raw data, different things can be concluded. First, the data could be used to derive user interests or contexts and how they change over time. The last fact is very important as user interests are not static. People change as they get more and more experience, a different job, etc. What is interesting for someone now, can be completely unhelpful later on, while other information could be useful again and again. Second, the data provides clues about how the user structures his data, which is the focus of our current research.

3. PERSONALIZED STRUCTURES

It has been shown, e.g. in [9], that structuring elements of a result list or, more general, elements of a document collection into a hierarchy of classes improves the user knowledge about the collection and where to find relevant documents. This has already been recognized and is implemented in different search engines like Vivisimo [4] and Exalead [2]. Vivisimo clusters the most relevant documents of a search by using document similarities. Exalead structures the result set by different aspects, e.g. language, document type, or domain. Furthermore, if result documents are part of the open directory [3], the result set is structure by its categories. However, different users would organize the same data differently because of their different perspectives. Therefore in contrast to these approaches, we are interested in personalizing a structured view on a collection.

As already said, we assume that the user already has structured a (small) part of the collection, e.g. by a bookmark or folder hierarchy. For the rest of the collection, we do not have any information about how the user would structure it. We are now interested in examining what can be learned about the user's structuring behavior and whether and how this can be applied to new unseen data (in the collection).

When structuring a collection, one has basically two possibilities, classification or clustering. Classification needs training examples for every possible class in order to assign a class to a new item, while clustering assumes no knowledge about class structures. However in our setting, we have training data for some classes, already known to the user. For the rest of the data, we know nothing about the class structure. Therefore, we have a scenario in between and can try to tackle the problem from two directions, either from the classification point of view (see Sec. 3.1) with integrating that our class knowledge is incomplete or from the clustering point of view (see Sec. 3.2) with integrating that we have some structural knowledge, both producing some kind of a semi-supervised learning approach.

3.1 A Classification Task

In [6, 7], we examined in first studies how the hierarchical structure provided by the user could be used to overcome the problem of unknown classes in a classification setting. For items that belong to one of the known classes, we want, of course, a prediction into that class. However, if we do not know the specific class of an item, we might already know that it belongs to a more general concept that we already have in our structure. Otherwise, we want to know that it does not fit into our structure at all.

Consider the following example. A user might be interested in machine learning algorithms, so he has different folders for different algorithms. For each algorithm, he again separates the documents, this time based on application fields. Furthermore, he is interested in Sports and has also different Sports folders. During his search, he encounters a document about "Neural Networks". As this is the first document about this topic, he does not have a folder about it yet. However, "Neural Networks" are a machine learning algorithm, and this higher level folder already exists. So the classifier could at least categorize the new document into this folder. Of course, there also will be documents, which even do not belong to higher level concepts and cannot be classified at all, so they should be assigned to the root level.

A problem that makes it harder to decide that "Neural Networks" should be assigned to the machine learning folder is that this folder might be completely empty as the user has only assigned documents to sub folders. So the question is how do we learn that a document should be assigned to the machine learning folder but not to one of its sub folders. There exist different approaches in the literature, e.g. [8, 13]. Our main idea is that we try to find the correct folder by using the uncertainty in the classification process. This means if the classifier produces almost the same values for two classes, it shows that the document does not clearly belong to one of the sub classes. However, it could clearly belong to a higher level class.

In our approaches presented in [6, 7], we therefore first train a standard classifier, in our experiments Naïve Bayes and SVM, with the available data on each node. For a new document, we predict class probabilities for each class by using the trained classifier. In a second step, we then reinterpret the classifier output to find the best fitting node in the hierarchy. As already said, the goal is a reduction of uncertainty which is achieved by generalizing the prediction in the class hierarchy. Of course, always predicting the most general class, the root node, reduces uncertainty at most. However, this is no more helpful to the user. So, we need to find a trade-off between uncertainty and specificity.

In [6], we selected the two best classes predicted by a Naïve Bayes classifier. The closer the classifier output for both classes P_1 and P_2 is the more uncertain is that the document really belongs to the best predicted class. However, we argue, if the two classes are already quite similar then of course the classifier output for these two classes will be also as these two classes are more difficult to distinguish. So instead of saying, if the difference in the classifier output is below a threshold, the classification is uncertain, we weight this threshold with the similarity of the two classes. If we actually decide that a classification is uncertain, we predict the class in the hierarchy, which is the most specific class that is parent of both compared classes. The algorithm can be found in Fig. 1.

1. Run NaiveBayes
2. sim = cosSim($class_1$, $class_2$)
3. If ($P_1 - P_2 <$ (1-sim)·C)
   a. Predict commonParent($class_1$, $class_2$)
   Else
   a. Predict $class_1$

Figure 1: Algorithm based on Class Similarity

In [7], we followed a hierarchical approach to reach the same goal. The algorithm starts in the root node and decides at each hierarchy level, whether to greedily descend into a sub class or stop descending at the current node, which is then returned as classification results. Instead of using the class similarity to gain some further knowledge about the compared classes, we use here the probabilities to predict a class correctly, which are determined in a previous step over the training data. In each step, the current node is compared with its best sub node. For more details, see [7].

An important question is how the performance of such a classification algorithm can be measured. Standard measures like precision and recall only measure exact matches. A classification is either correct or wrong. However, for a hierarchy of classes this is not completely true. As we argue in [7], a classification into a class that is a parent class of the correct class is still helpful to a user although not as specific as it could have been. Therefore classification in such a node should still be counted as "partly" correct. For this purpose, we suggest an adapted precision and recall that counts classification in a parent node depending on the distance to the actual node (see Eq. 1-3).

$$prec_c = \frac{\sum_{D \in retrieved_c} \text{sim}_{\text{tree}}(C, \text{class}(D))}{|retrieved_c|} \quad (1)$$

$$recall_c = \frac{\sum_{D \in relevant_c} \text{sim}_{\text{tree}}(\text{predClass}(D), C)}{|relevant_c|} \quad (2)$$

$$\text{sim}_{\text{tree}}(C_{pred}, C_{cor}) = \begin{cases} 1/\left(\text{dist}_{\text{tree}}(C_{pred}, C_{cor}) + 1\right) & \text{if } C_{pred} \geq_h C_{cor} \\ 0 & \text{else} \end{cases} \quad (3)$$

Other researchers also proposed hierarchical performance measures, e.g. in [13]. However, their focus is more on evaluating the classifier performance itself, e.g. by taking category similarities or tree distances into account. Our focus is on the usefulness of the classification from a user's point of view, which is based on user behavior in the described application scenario.

3.2 A Clustering Task

In our current work, we examine how automatic grouping methods can be applied. That means our starting point here is a clustering algorithm, into which we want to integrate the knowledge about known personal structures. As basis, we use a hierarchical agglomerative clustering algorithm (HAC) as this enables us to learn a hierarchical structure of the data. Related work, e.g. [15], usually works on flat clustering techniques. The items already organized by the user are mixed together with the unclassified result set items and the complete set is clustered. However, we have to make sure that the items are grouped in a way the user would have done it, which need not be true for standard clustering algorithms. That means the known documents should be clustered as in the user's document hierarchy. We are currently studying how modifications of the term influence in the similarity measure can serve this goal. Furthermore, we are trying to find performance measures that can be used to reflect the personalized clustering perspective.

3.3 Integrating Semantics

For future work, we also want to integrate semantics in order to learn more about the user's general organizational behavior. Let's come back to our example user who is interested in machine learning. If this user looks for a new algorithm, he most likely would want to organize it in his machine learning folder in a new sub folder titled as the algorithm with sub folders describing applications. For doing this automatically, it is probably not enough to only work with extracted feature vectors to describe documents and categories. Further knowledge about the meaning of single terms is necessary. With a named entity recognizer [10], special entities could be extracted. Ontologies could be used to derive the semantics of a term, e.g. that "Neural Network" is a machine learning algorithm. Furthermore, we need to look on other features like some users might prefer alphabetic ordering, others might order their items by time. The same techniques can also help when we want to find good and useful labels for new classes generated.

4. USER GROUPS

Using personal user data has the advantage that this data is specific to this user. However, the amount of data available for a certain user is usually rather small, compared to the huge amount of data available in the collections. An approach to resolve this issue is the formation of user groups. Here, the collected data of a group of users is combined to form one large profile. However, we argue that especially when interacting with such heterogeneous mediums like the web, no one matches in all aspects with somebody else, although that might be true for certain aspects. E.g. with some users I share my work interests, with others I share my favorite country to make holiday in, and again with others I share my hobby. Therefore, instead of putting users entirely together, we prefer doing it on the basis of interests or contexts. Most currently applied collaborative or social filtering techniques still consider each person as an entity. Furthermore, for a certain interest, the integration of further knowledge from other users sharing the same interest can be very helpful to enrich the result visualization, the derived structure, and the labels created for found documents.

The question that remains is how such a user group can be formed. An automated approach could work on the previously mentioned learned user interests or contexts derived from the user profile. For each interest of a user, an algorithm could look for similar interests in other user profiles. The data specific to this interest from other users can then be used to enrich exactly this interest, while data from a different set of users enriches another interest of the user. On the other hand, users could also join a group manually for certain interests by subscribing to social communities that share a common interest. This is, for example, implemented in the search of Eurekster [1]. Here, a user issues a search inside a search group about a specific topic. Users can create new search groups or join existing ones.

5. RESULT VISUALIZATION

Another important question is how the personalized results should be visualized to the user. When displaying data in a hierarchical class structure, the labels of each class are essential in order for the user to be able to navigate efficiently through the hierarchy. Automatic extraction of useful labels is still an open research issue, even though, labeling based on named entities works well for many applications, see e.g. [14]. However, we argue that class names chosen by the user or group members could help to improve labeling as the manually chosen labels are probably quite expressive. This might even be true for parts of the collection that do not belong to any interest of the user. Here, the profile of other users can help, who already have structured the data in their profiles.

Furthermore, special care should be taken to differentiate between finding new information and re-finding previously seen data as people use search engines for both purposes. Pages that were already visited by the user should be highlighted as such. However, further information like the date of the last access is important in order for the user to know whether this document is probably long forgotten or just the one from yesterday. It is probably also interesting to know, whether the document changed since the last visit. It might also help the user, if the visualization includes information to state what items belong to profiles of other group members, as this might suggest stronger importance of the document.

For search queries, it could also be helpful to visualize already performed searches with the same or similar keywords. For re-finding, it is especially of interest, what the results of these searches were, this is which pages were accessed as a consequence of the search. And this does not limit to direct search results but also includes links that were followed further on. For finding new information, this helps to avoid revisiting already known items, which could save precious time.

Besides approaches for personalized visualization of document collections [11], we study in related projects recommendation methods for browsing collections, especially the Web [12]. Here, it is important to visualize recommended links and further information about them. All projects examine different aspects of the retrieval and personalization process and we are currently working towards more integrated solutions.

With all the data available, an interface must be carefully designed in order to avoid information overload. Also here, personalization is important. That means the user should get complete control to decide on what he wants to see. This includes the kinds of information to display as well as how information is displayed. Concerning the hierarchical structure, the number of classes per level, the depth of the hierarchy or the number of documents to display per class are surely criteria that should be user controlled and might also depend on the context of the search.

6. CONCLUSION

In this paper, we presented our ideas and work regarding personal information management. Our main focus is on the creation of a personalized structured view on a collection or a subset of it (like a search result list). Herein, we are interested in examine what can be learned about a user's structuring behavior and how this can be applied to new data. Another important aspect of our work is to derive measures for evaluating the performance increase by the personalization to reduce the need for user studies. We presented our first studies based on a classification task together with an appropriate evaluation measure and introduced our current work and future plans. Furthermore, we presented our ideas about the positive influence of group data and visualization to further improve the view on the collection. All algorithms are integrated in our CARSA system [5], which gives us a general framework for conducting extensive tests and later on also for user studies.

7. REFERENCES

[1] Eurekster search. http://search.eurekster.com, 2006.

[2] Exalead search engine. http://www.exalead.com, 2006.

[3] Open directory project. http://www.dmoz.org, 2006.

[4] Vivisimo search. http://www.vivisimo.com, 2006.

[5] K. Bade, E. De Luca, A. Nürnberger, and S. Stober. CARSA - an architecture for the development of context adaptive retrieval systems. In *Proc. of the 3rd Intern. Workshop on Adaptive Multimedia Retrieval*.

[6] K. Bade and A. Nürnberger. Supporting web search by user specific document categorization: Intelligent bookmarks. In *Proc. of 13. Leipziger Informatik-Tage (LIT 2005)*, pages 115 – 123, 2005.

[7] K. Bade and A. Nürnberger. Rearranging classified items in hierarchies using categorization uncertainty. In *Proc. of 30th Annual Conference of the German Classification Society*. Springer, 2006. (to appear).

[8] B. Choi and X. Peng. Dynamic and hierarchical classification of web pages. *Online Information Review*, 28(2):139–147, 2004.

[9] S. T. Dumais, E. Cutrell, and H. Chen. Optimizing search by showing results in context. In *Proc. of the 2001 Conference on Human Factors in Computing Systems*, pages 277–284, 2001.

[10] H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proc. of COLING-2002*, pages 390–396, 2002.

[11] A. Nürnberger and M. Detyniecki. Weighted self-organizing maps: Incorporating user feedback. In *ICANN: International Conference on Artificial Neural Networks*, pages 883–890, 2003.

[12] S. Stober and A. Nürnberger. Context-based navigational support in hypermedia. In *Proc. of the 2006 Intern. Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, 2006. (to appear).

[13] A. Sun and E. Lim. Hierarchical text classification and evaluation. In *Proc. of the 2001 IEEE International Conference on Data Mining*, pages 521–528, 2001.

[14] H. Toda, M. Nagahama, H. Nakawatase, and R. Kataoka. A label-based navigation method using informatively named entities. In *Proc. of the First International Workshop on Knowledge Discovery in Data Streams*, 2004.

[15] S. Zhong. Semi-supervised model-based document clustering: A comparative study. *Machine Learning*, 2006.