Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence, pp. 391-398, Chicago, November 1999

Ontology Based Personalized Search[*]

Alexander Pretschner[†]
Institut für Informatik
Technische Universität München
Arcisstraße 21, D-80290 München, Germany
`pretschn@in.tum.de`

Susan Gauch
Department of EECS
The University of Kansas
415 Snow Hall, Lawrence, KS 66045, USA
`sgauch@eecs.ukans.edu`

Abstract

With the exponentially growing amount of information available on the Internet, the task of retrieving documents of interest has become increasingly difficult. Search engines usually return more than 1,500 results per query, yet out of the top twenty results, only one half turn out to be relevant to the user. One reason for this is that Web queries are in general very short and give an incomplete specification of individual users' information needs. This paper explores ways of incorporating users' interests into the search process to improve the results. The user profiles are structured as a concept hierarchy of 4,400 nodes. These are populated by 'watching over a user's shoulder' while he is surfing. No explicit feedback is necessary. The profiles are shown to converge and to reflect the actual interests quite well. One possible deployment of the profiles is investigated: re-ranking and filtering search results. Increases in performance are moderate but noticeable and show that fully automatic creation of large hierarchical user profiles is possible.

1. Introduction

As of March 1999, the Internet provides about 165 million users worldwide with every imaginable type of information (source: Nua Internet Surveys, www.nua.ie/surveys). In general, people have two ways to find the data they are looking for: they can *search*, and they can *browse*. Search engines index millions of documents on the Internet and allow users to enter keywords to retrieve documents that contain these keywords. Browsing is usually done by clicking through a hierarchy of subjects until the area of interest has been reached. The corresponding node then provides the user with links to related websites. The search and browsing algorithms are essentially the same for all users.

It is unlikely that 165 million people are so similar in their interests that one approach to searching or browsing, respectively, fits all needs. Indeed, in terms of searching, about one half of all retrieved documents have been reported to be irrelevant [3]. The main problem is that there is too much information available, and that keywords are not always an appropriate means of locating the information in which a user is interested. Presumably, information retrieval will be more effective if individual users' idiosyncrasies are taken into account. This way, an effective personalization system could decide autonomously whether or not a user is interested in a specific webpage and, in the negative case, prevent it from being displayed. Or, the system could navigate through the Web on its own and notify the user if it found a page or site of presumed interest.

This paper studies ways to model a user's interests and shows how these models - also called profiles - can be deployed for more effective information retrieval and filtering. A system is developed that "watches over the shoulder" of a user while he is surfing the Web. A user profile is created over time by analyzing surfed pages to identify their content and by associating that content with the length of the document and the time that was spent on it. When pages about certain subjects are visited again and again, this is taken as an indication of the user's interest in that subject. Except for the act of surfing, no user interaction with this system is necessary. This paper shows how the profiles can be used to achieve search performance improvements. The increases in performance are modest, but they are noticeable, and they are a first step.

This work has been carried out as part of the OBIWAN project (www.ittc.ukans.edu/obiwan, [30]) at the University

---

[*] The research presented in this paper was partially supported by the National Science Foundation CAREER Award number 97-03307. The first author was in part supported by the German-American Fulbright Program.

[†] This work was carried out while the first author was at the University of Kansas.

of Kansas. The goal of OBIWAN is to investigate a novel content-based approach to distributed information retrieval. Websites are clustered into regions. Examples for clustering criteria include but are not restricted to content, geographic location, and association with a specific company. Regions are clustered into super regions, super regions into hyper regions, etc., thus forming a hierarchy of regions. A node of this hierarchy can be browsed by simultaneously browsing its child nodes. In terms of searching, queries are brokered within one node by deciding which child nodes are the most promising candidates for the retrieval process. This is done by determining the content of the query and using a sitemap containing information about the content of every node in the (sub)hierarchy: the query is brokered to those nodes with a content that best matches the content of the query. The results of the child nodes are then merged and returned to the parent node or, eventually, to the initiator of the query. The text classifier, which is described in section 3, is a core component not only of the entire OBIWAN project, but also of the presented personalization method.

2. Related Work

Personalization is a broad field of very active ongoing research. Applications include *personalized access* to certain resources (e.g., personalized "portals" to the web, e.g. MyYahoo, FireFly or PointCast at www.yahoo.com, www.firefly.net, and www.pointcast.com, respectively) and filtering/rating systems: electronic newspapers (e.g., Wall Street Journal or FishWrap [4] at www.wsj.com and www.sfgate.com, respectively), Usenet news filtering, recommendation services (browsing, navigation), and search. [20] describes about 45 personalization systems and contains a detailed bibliography.

To the authors' knowledge, SmartPush [9] is currently the only system to store profiles as concept hierarchies. These are much smaller (40-600 nodes), and weight adjustments are done with respect to data that *explicitly* describes the document contents. It is doubtful that hand-made hierarchical content annotation – i.e., not just lists of keywords as in the case of XML – of data will be done on a large scale. Systems that use structured information rather than simple lists of keywords include PEA [15] and SiteSeer [21] (bookmark structure), PSUN [25] (K-lines), and SiteIF [26] (semantic networks).

Browsing behavior is used for data acquisition in Anatagonomy [22], Letizia [12, 11], Krakatoa [7], Personal WebWatcher [14], and WBI [2]. Usenet news filtering systems include GroupLens [8], PSUN [25], NewT [24], and SIFT [28]. In addition to news filtering, Amalthaea [16] explores (autonomous) personalized data discovery on the Web. SiteIF [26] and ifWeb [1] aim at personalized search and navigation support. Syskill and Webert [19] is another example of a personalized recommendation service. InformationLens [13] is a tool for filtering and ranking e-mails. Finally, [27] describes a system for expertise location (JAVA source codes). [20] contains a thorough discussion of these and other systems. Implicit rating and filtering are, among others, discussed in [17] and [18].

3. Determining the content of documents

User interests are inferred by analyzing the web pages the user visits. For this purpose, it is necessary to determine the content, or characterize, these surfed pages. This is done by using a hierarchy of concepts, or rather ontolo[illegible] This ontology is based on a publicly accessible browsing hierarchy. For this paper, the Magellan hierarchy, which is comprised of approximately 4,400 nodes, has been mirrored (magellan.excite.com). The nodes of the ontology are labelled with the names of the nodes in the browsing hierarchy. The semantics of the edges of this hierarchy is not formally specified; in most cases, they correspond to a specialization relation (super-/subconcept).

Each node of the browsing hierarchy is associated with a set of documents that are used to represent the content of this node. All of the documents for a node (in the experiments, 10 documents per node) are merged into a superdocument. Documents as well as superdocuments are represented as weighted keyword vectors using the vector space model [23]. The weights are based on term frequencies and inverted document frequencies: It is assumed that multiple occurences of a word indicate that its meaning contributes to the content of the document more than that of less frequent terms. However, words that occur with a very high overall frequency (i.e., in the collection of documents in question) do not discriminate between documents within this collection.

For each of the surfed pages a keyword vector is calculated. This page vector is compared with the keyword vectors associated with every node to calculate similarities. The nodes with the top matching vectors are assumed to be most related to the content of the surfed page. The accuracy of this text categorization algorithm was validated in [30].

4. User Profiles

User profiles store approximations of the interests of a given user. The proposed generation of user profiles differs from the majority of other approaches in that the profiles are

1. hierarchically structured, and not just a list of keywords,

2. generated automatically, without explicit user feedback, and

3. dynamic, i.e., the learning process does not necessarily stop after a fixed period of time.

4.1. Creation and Maintenance

Profiles are generated by analyzing the surfing behavior of a user. “Surfing behavior” here refers to the length of the visited pages and the time spent thereon. No user feedback is necessary. It is the authors’ belief that a system with an explicit feedback mechanism does not encourage the user to deploy such a system – even if a simple assessment “relevant” or “non-relevant” does not take more than a second, it considerably disrupts the user’s workflow and is hence annoying.

The profile generation and adaptation work as follows: The files in a web browser’s cache folder are periodically characterized, i.e., subject areas, or categories, are assigned to each page. The strengths of match for the top five categories are then combined with the time a user spent on the page and the length of that page. This yields an update value for the five categories. Currently, weights can only increase: no attempt is made to infer whether or not a user *dis*liked a page and the associated categories from their browsing behavior.

Four different combinations of time, length, and subject discriminators have been investigated. In the following discussion, *time* refers to the time a user spent on a given page, and *length* refers to the length of the page (i.e., the number of characters). Let $\gamma(d, c_i)$ be the strength of the match between the content of document d and category c_i. This value is a result of the characterization process of a page. The adjustment of the interest ι in a category c_i, $\iota(c_i)$, will be denoted by $\Delta\iota(c_i)$. In terms of convergence and search result improvements (see below), two functions have shown to be superior to the other two. These superior functions are $\Delta\iota(c_i) = \log \frac{time}{\log length} \cdot \gamma(d, c_i)$ and $\Delta\iota(c_i) = \log \frac{time}{\log \log length} \cdot \gamma(d, c_i)$. They share the commonality of not heavily taking into account the length of a page.[1] In practice, these measures are modified to guarantee a positive interest value.

4.2. Profile Evaluation: Convergence

The evaluation of the user profiles consists of two parts. First, a notion of convergence is introduced with respect to which 16 actual user profiles are discussed. This relatively small number of experiments is due to the fact that users seem to be well aware of privacy issues and are rather reluctant to allow others to access their surfing history. The second part examines the relationship between the calculated user interests and the actual user interests. For the experiments, a group of 16 users were monitored for 26 days. These 16 users together surfed 7,664 documents (which may contain double counts). The mean time spent was $\mu = 54.6\,\mathrm{sec}$; median $\tilde{x} = 18\,\mathrm{sec}$, the standard deviation of $\sigma = 93.4$ is rather large. 20% of all pages are visited for less than $5\,\mathrm{sec}$ (see [20] for details).

One would assume that each person has a relatively stable collection of interests which may change over time [10]. Thus, the evaluation of the profiles will be based on a notion of convergence. In the following, a *node* always refers to a node in the subject hierarchy and hence to a category or a subject. A user profile is said to be *convergent* if the number of nodes with non-zero interest values ι converges over time. Note that this is not a technical definition since the notion of “convergence” of a set of values is not specified. Figure 1 shows a sample profile (adjustment function $\Delta\iota(c_i) = \log \frac{time}{\log(\log length)} \cdot \gamma(d, c_i)$). It consists of roughly 75 non-zero categories.

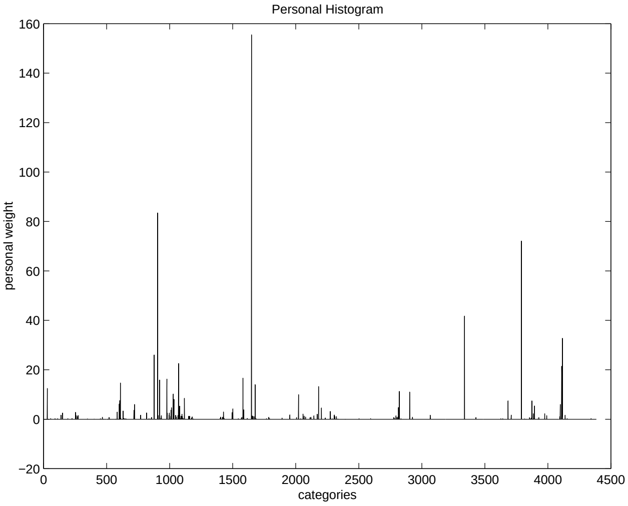

Figure 1. Sample user profile: less than 100 categories. Categories are numbered sequentially.

Users varied in the number of categories their profiles converged to, most falling between 50 and 200. Figure 2 shows the numbers of non-zero categories for five sample profiles with 100-150 categories created using the same interest adjustment function. It is possible for the total number of categories to decrease since actually, the numbers do not represent non-zero categories but rather the number of categories accounting for 95% of the total accumulated personal weight. This is done to filter “noise” that is introduced by inaccuracies of the text classifier.

---

[1] The functions that yielded poor results in terms of convergence and search result improvements were $\Delta\iota(c_i) = \frac{time}{length} \cdot \gamma(d, c_i)$ and $\Delta\iota(c_i) = \log \frac{time}{length} \cdot \gamma(d, c_i)$.

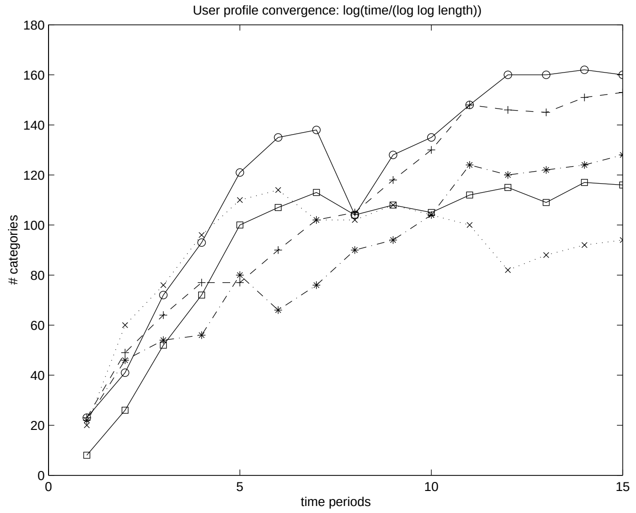

Figure 2. Convergence of five profiles with less than 150 categories

The time intervals in Figure 2 are actually not clock time but rather represent periods of activity in which an equal number of documents (on average, about 20) have been surfed. In this way, idle times like weekends or vacations do not confuse the overall image, and the evaluation is consistent between users who surf at different times.

With the two aforementioned interest adjustment function, all profiles show a tendency to converge after roughly two thirds of all documents have been surfed: The curves eventually become “flatter” after ten units on the x-axis. On average, that corresponds to roughly 320 pages, or 17 days of surfing. Table 1 summarizes the convergence properties (numbers have been determined graphically). In terms of profile convergence, both functions seem to be equally well suited.

Table 1. Convergence of interest adjustment functions

function	average units for convergence
$\frac{time}{length}$	no convergence
$\log \frac{time}{length}$	no convergence
$\log \frac{time}{\log length}$	9.6
$\log \frac{time}{\log \log length}$	9.4

With respect to convergent behavior, this also indicates that the length of a surfed page does not really matter in determining the interest in it. The goal of incorporating the length into the above formulae was to normalize time by length. This seems unnecessary because users can tell at a glance that a page is irrelevant and, in general, reject it quickly, regardless of its length. There seems to be no need for normalization because if they spend longer on a page, it is because it is relevant, and not because it is longer.

Table 2. Profiles vs. actual interests for 20 and 10 categories (lower part). n=16.

	how many good ones	how well subset	how well all	how many bad ones	bad/ good
μ	10.5	3.7	2.8	5.3	1:2
σ	4.8	1.0	1.0	5.3	-
$\tilde{x}$	9	4	3	3	-
μ	5.2	3.5	2.5	3	1:1.7
σ	2.3	1.0	1.4	2.4	-
$\tilde{x}$	5	4	3	2	-

4.3. Comparison with actual user interests

Although convergence is a desirable property, it does not measure the accuracy of the generated profiles. Thus, the sixteen users were shown the top twenty subjects in their profiles in random order and asked how appropriately these inferred categories reflected their interests:

1. *How many* of the above 20 subjects do reflect your actual interests?
2. *How well* does that subset (i.e., the subjects describing your interests) reflect your actual interests (0=very bad . . . 5=very good)?
3. *How well* does the entire set of 20 categories describe your actual interests (0=very bad . . . 5=very good)?
4. *How many* of the above subjects do not reflect your interests at all?
5. Please answer questions 1-4 by only looking at the top 10 categories, i.e., discard the second half of the list!

Table 2 shows mean μ, standard deviation σ, and median $\tilde{x}$ for the answers to the above questions with the top 20 and top 10 categories, respectively ($n = 16$), for one of the better interest adjustment functions. In both cases, approximately one half of the categories represent actual interests. The reason for this is most likely the suboptimal accuracy of the categorization algorithm. Bearing in mind that the “good” categories have been chosen out of as many as 4,400 categories, this result is still surprisingly accurate. One half of 20 categories chosen reflect actual interests even though these represent only 0.5% of all possible categories.

If emphasis is put on these “good” categories, users feel represented well - a value of 3.5 might be verbalized as “pretty good”. Since roughly one half of the categories do not represent user interests, it is not surprising that the entire set does neither represent nor misrepresent actual interests. Finally, only a quarter to a third of all the categories do not represent interests at all. (There is a difference between “not representing at all” and “not representing well”.) The goal of the next section is to evaluate whether this qualitative feedback translates into quantitative improvements for some task (in this case, re-ranking and filtering).

5. Improving Search Results

The wealth of information available on the web is actually too large: when entering a query into a search engine such as AltaVista, too many results are retrieved. The number of results regularly exceeds 1,500, and the top ranked documents a user can have a look at are often not relevant to this user. This happens due to an inherent problem in keyword based search: search terms are ambiguous; their meaning depends on the context and, more importantly, on the meaning a user assigns to them.

In the evaluation of the proposed system, 48 query results have been judged by 16 users, the judgment being either “relevant” or “irrelevant”. On average, only $\mu = 8.7$ out of 20 result pages were considered to be relevant (median $\tilde{x} = 8.5$, standard deviation $\sigma = 3.0$). This is consistent with the findings in [3] which reports that roughly 50% of the retrieved documents are irrelevant (with a statistically more significant set of 1,425 queries and 27,598 judged results).

There are three common approaches to address this problem:

- Re-Ranking
  Re-Ranking algorithms apply a function to the ranking numbers that have been returned by the search engine. If that function is well chosen, it will bring more relevant documents to the top of the list.

- Filtering
  Filtering systems determine which documents in the results sets are relevant and which are not. This is usually done by comparing the documents to a list of keywords that describe a user or a set of documents that the user previously judged relevant or irrelevant, respectively. Good filters filter many non-relevant documents and do keep the relevant ones in the results set.

- Query Expansion
  Often, queries are very broad. If a query can be expanded with the user’s interests, the search results are likely to be more narrowly focused. However, this is a very difficult task since query reformulating needs to expand the query with relevant terms. If the expansion terms are not chosen appropriately, even more irrelevant documents will be returned to the user.

This section uses the profiles of the previous section to implement the first two approaches.

5.1. Re-Ranking

Given a query, re-ranking is done by modifying the ranking that was returned by a publicly accessible search engine, ProFusion (www.profusion.com) in this case. The idea is to characterize each of the returned documents (or rather their title together with their summary, which, according to [3] and [19] is sufficient for classification purposes) and, by referring to the user profiles, to determine how much a user is interested in these categories. The user’s average interest in the document’s top categories is assumed to be an approximation to the actual user interest in the whole document.

Remember that $\gamma(d, c_i)$ denotes a measurement of how well category c_i describes the content of a document d. Let $\pi(c_1) \ldots \pi(c_4)$ be the personal interests assigned to the top four categories $c_1 \ldots c_4$. d_j denotes the documents as returned by ProFusion ($1 \leq j \leq 20$), and $w(d_j)$ denotes the rank value that ProFusion assigned to these documents. Five re-ranking functions have been evaluated. They are all similar to $\varrho(d_j) = w(d_j) \cdot \left(.5 + \frac{1}{4} \sum_{i=1}^{4} \pi(c_i) \cdot \gamma(d_j, c_i) \right)$ in that the multiplication is replaced by a weighted sum. Furthermore, it was necessary to normalize the personal interests.

5.2. Evaluation

The results that have been produced by the different re-ranking systems must be evaluated. Since these results are in the form of rank-ordered URLs, it is necessary to select an objective measure for the relative quality of two rank-ordered lists. The eleven point precision average [6] is one such measure. The basic idea is to cluster documents into two groups, the relevant and the non-relevant ones and to check how many relevant documents appear at the top of the re-ranked list. This measure has one disadvantage in that it considers all relevant documents to be equally relevant. The n-dpm (normalized distance based performance measure [29]) overcomes this restriction; [20] evaluates the presented system in terms of it.

The eleven point precision average evaluates ranking performance in terms of *recall* and *precision*. Recall is a measure of the ability of the system to present all relevant items, and precision is a measure of the ability of a system to present only relevant items: $\text{recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$, and

$\text{precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}$. A system's performance can be described by relating eleven interpolated recall cutoffs with their respective precisions. By averaging over the uninterpolated values (on a per-query basis), the system performance change can be measured by one single value.

16 users were asked to submit three queries. The results were presented to them in random order, and they were asked to judge each result as being "relevant" or "non-relevant". (For evaluation in terms of the n-dpm, they were also asked to actually rank the relevant results. This is why only three queries per user were chosen.) The $16 * 3 = 48$ queries were partitioned into a training set of 32 documents and a testing set of 16. Figure 3 shows the recall-precision graphs for one interest adjustment functions. Those curves above the "ProFusion" curve exhibit better system performance than ProFusion itself. The curves correspond to the re-ranking functions that have been mentioned above (multiplication and different weighted sums). For instance, "more ProFusion" means that in the weighted sum, the original ranking is weighted three times as high as the personalized contribution. According to this figure, the multiplicative ranking function exhibits the best performance increase (up to 8%). The remaining set of 16 queries were evaluated using this function (figure 4); the findings are consistent.

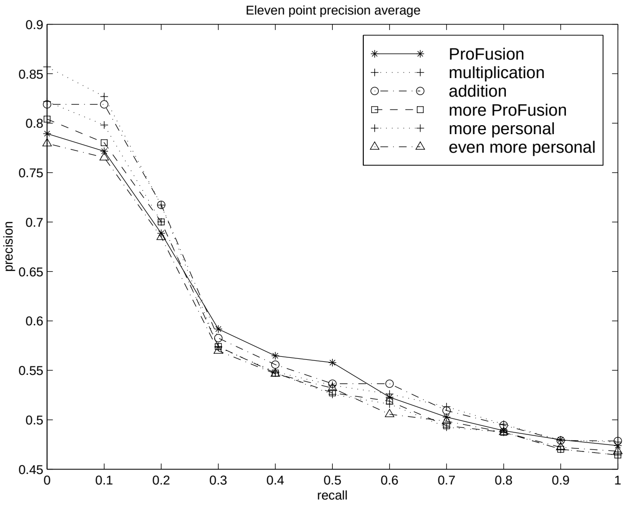

Figure 3. 11 point average precision for interest adjustment function $\Delta\iota(c_i) = \log\frac{time}{\log(\log length)}\cdot\gamma(d, c_i)$ and 5 different ranking formulae.

The same set of experiments were conducted with the normalized distance-based performance measure. The same two interest adjustment functions were the only ones to increase system performance. This increase summed up to 3% [20].

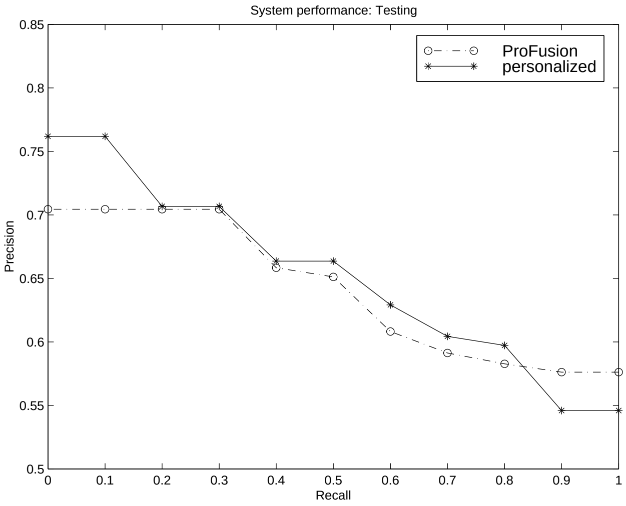

Figure 4. 11 point average for the testing set: 16 queries

5.3. Filtering

To filter a set of result documents means to exclude some (hopefully irrelevant) documents. Filtering was done by using the above ranking functions with thresholds to decide which documents were irrelevant and which were not. It turned out that again, the same two interest adjustment functions resulted in performance increases. Figures 5 and 6 show the performance of the filter for the training and the testing set, respectively.

The figures indicates that, for large threshold values, there are two to three times more irrelevant than relevant documents filtered. However, one should note that the absolute number (7% of 20, or 1.4 documents per query for a threshold value greater than 0.8) is rather small.

Although the filter improves search results, these improvements are modest (9-15% with 6-12% incorrectly filtered documents). This suggests that the system performs better in ranking than in filtering. This is likely due to the fact that the decision "relevant" vs. "non-relevant" is very coarse and that mistakes are easily made. In the case of re-ranking, switching the position of two items does, in general, not greatly affect the quality of the results. Explicit user feedback may be necessary to achieve high-quality filtering.

6. Conclusions and Future Work

A system has been presented that allows for the fully automatic creation of large structured user profiles. These profiles have been shown to converge and to reflect actual

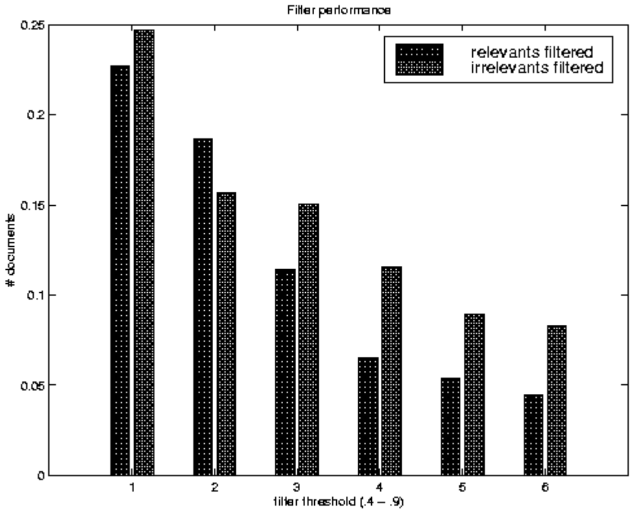

Figure 5. Average filter performance: Training. Adjustment function $\Delta\iota(c_i) = \log\frac{time}{\log length}\cdot\gamma(d, c_i)$. Abscissa: threshold values from .4 to .9; ordinate: ratio of relevant (non-relevant) documents.

user interests quite well. To evaluate their usability, two applications have been investigated: Re-ranking and filtering search results. In terms of re-ranking, performance increases of up to 8% have been detected (11 point average measure). In terms of filtering, the results were more moderate (9-15% irrelevant and 6-12% relevant documents filtered).

With the presented approach, the length of a surfed page can be neglected when the interest in a page is inferred. What matters more, is the time spent on that page.

Future work includes the integration of the system into a web browser (right now, cache folders are analyzed) which will allow for more accurate interest detection if other interactions such as scrolling behavior are monitored. Other ideas include personalizing the structure of the ontology by splitting or coalescing nodes. Furthermore, it seems possible to (unsupervisedly) re-train the text classification algorithm.

In addition, other areas of profile deployment are conceivable. These include expertise location and recommendation services, e.g., for books. In terms of privacy, the existing system stores the profile on the user's machine. Since search results are post processed, there is no need for a central profile server. However, for the mentioned recommendation services or expertise location, ways of protecting the profiles have to be investigated (e.g., [5]).

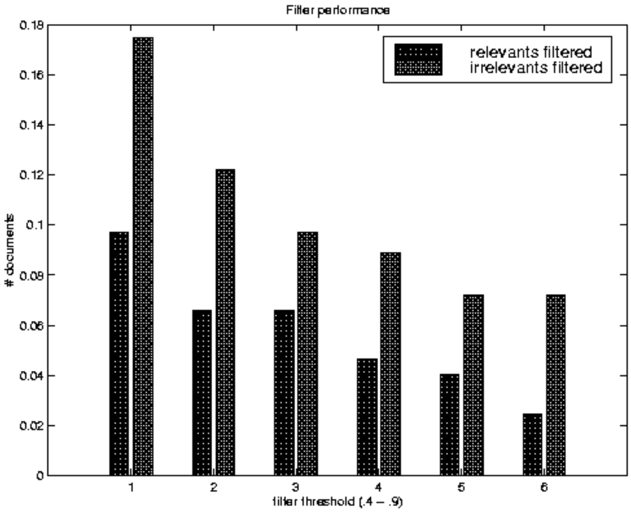

Figure 6. Average filter performance: Testing. Adjustment function $\Delta\iota(c_i) = \log\frac{time}{\log length}\cdot\gamma(d, c_i)$. Abscissa: threshold values from .4 to .9; ordinate: ratio of relevant (non-relevant) documents.

References

[1] F. Asnicar and C. Tasso. ifWeb: a prototype of user model-based intelligent agent for documentation filtering and navigation in the World Wide Web. In *Proc. 6th Intl. Conf. on User Modeling,* Chia Laguna, Sardinia, June 1997.

[2] R. Barrett, P. Maglio, and D. Kellem. How to personalize the Web. In *Proc. ACM CHI'97,* Atlanta, USA, 1997.

[3] E. Casasola. ProFusion PersonalAssistant: an agent for personalized information filtering on the WWW. Master's thesis, The University of Kansas, Lawrence, KS, 1998.

[4] P. Chesnais, M. Mucklo, and J. Sheena. The fishwrap personalized news system. In *Proc. IEEE 2nd Intl. Workshop on Community Networking Integrating Multimedia Services to the Home,* Princeton, New Jersey, USA, June 1995. http://fishwrap.mit.edu.

[5] E. Gabber, P. Gibbons, D. Kristol, Y. Matias, and A. Mayer. Consistent, yet anonymous web access with LPWA. *Communications of the ACM,* 42(2):42–47, February 1999.

[6] D. Harman. Evaluation Techniques and Measures. In *Proc. 4th Text Retrieval Conference (TREC-4),* pages A–6–A–14, October 1996.

[7] T. Kamba, K. Bharat, and M. Albers. The Krakatoa Chronicle - an interactive, personalized newspaper on the Web. In *Proc. 4th Intl. WWW Conf.,* pages 159–170, 1995.

[8] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM,* 40(3):77–87, March 1997.

[9] T. Kurki, S. Jokela, R. Sulonen, and M. Turpeinen. Agents in delivering personalized content based on semantic metadata.

In *Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 84–93, Stanford, USA, 1999.

[10] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proc. ACM SIGIR'96*, Zurich, Switzerland, 1996.

[11] H. Liebermann. Letizia: An agent that assists Web browsing. In *Proc. Intl. Conf. on AI*, Montréal, Canada, August 1995.

[12] H. Liebermann. Autonomous interface agents. In *Proc. ACM Conf. on Computers and Human Interaction (CHI'97)*, Atlanta, USA, May 1997.

[13] T. Malone, K. Grant, F. Turbak, S. Brobst, and M. Cohen. Intelligent information sharing systems. *Communications of the ACM*, (30):390–402, May 1987.

[14] D. Mladenić. Personal WebWatcher: design and implementation. Technical Report IJS-DP-7472, J. Stefan Institute, Department for Intelligent Systems, Ljubljana, 1998.

[15] M. Montebello, W. Gray, and S. Hurley. A personable evolvable advisor for WWW knowledge-based systems. In *Proc. 1998 Intl. Database Engineering and Application Symposium (IDEAS'98)*, pages 224–233, Cardiff, Wales, UK, July 1998.

[16] A. Moukas. Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. In *Proc. 1st Intl. Conf. on the Practical Application of Intelligent Agents and Multi Agent Technology*, London, 1996.

[17] D. Nichols. Implicit rating and filtering. In *Proc. 5th DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary, November 1997. ISBN 2-912335-04-3.

[18] D. Oard and G. Marchionini. A conceptual framework for text filtering. Technical Report EE-TR-96-25 CAR-TR-830 CLIS-TR-9602 CS-TR-3643, University of Maryland, May 1996.

[19] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill&Webert: identifying interesting web sites. In *Proc. 13th Natl. Conf. on Artificial Intelligence*, 1996.

[20] A. Pretschner. Ontology Based Personalized Search. Master's thesis, The University of Kansas, Lawrence, KS, 1999. www4.in.tum.de/∼pretschn/papers/kuthesis.ps.gz.

[21] J. Rucker and M. Polanco. Siteseer: Personalized navigation for the web. *Communications of the ACM*, 40(3):73–75, 1997.

[22] H. Sakagami and T. Kamba. Learning personal preferences on online newspaper articles from user behaviors. In *Proc. 6th Intl. World Wide Web Conf.*, pages 291–300, 1997.

[23] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989. ISBN 0-201-12227-8.

[24] B. Sheth. A learning approach to personalized information filtering. Master's thesis, Massachusetts Institute of Technology, February 1994.

[25] H. Sorensen and M. McElligott. PSUN: a profiling system for Usenet news. In *Proc. CIKM'95 workshop on Intelligent Information Agents Workshop*, Baltimore, USA, December 1995.

[26] A. Stefani and C. Strappavara. Personalizing access to web sites: The SiteIF project. In *Proc. 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98*, Pittsburgh, USA, June 1998.

[27] A. Vivacqua. Agents for expertise location. In *Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 9–13, Stanford, USA, 1999.

[28] T. Yan and H. Garcia-Molina. SIFT - a tool for wide-area information dissemination. In *Proc. 1995 USENIX Technical Conf.*, pages 177–186, 1995.

[29] Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *J. of the American Society for Information Science*, 46(2):133–145, 1995.

[30] X. Zhu, S. Gauch, L. Gerhard, N. Kral, and A. Pretschner. Ontology based web site mapping for information exploration. In *Proc. 8th Intl. Conf. on Information and Knowledge Management (CIKM'99), to appear*, Kansas City, MO, USA, November 1999. http://www.ittc.ukans.edu/obiwan/publications/papers/CIKM.html.