

Optimal link categorization for minimal retrieval effort* †

Vera Hollink
Faculty of Science
University of Amsterdam
Amsterdam, The Netherlands
vhollink@science.uva.nl

Maarten van Someren
Faculty of Science
University of Amsterdam
Amsterdam, The Netherlands
maarten@science.uva.nl

ABSTRACT

Various studies have shown that categorizing search results can help users to retrieve their target pages faster. The categorizations save the users the time needed to consider links from irrelevant categories. However, what is often missed out is that the category selections also introduce extra effort for users who are looking for one of the highest ranked results. In general, the expected gain of categorization depends on the relative probability that the user is looking for each of the search results. In this work we present a method to balance the costs of presenting categories against the expected savings of the categorization. In an experiment we demonstrate that this method can reduce retrieval time substantially compared to flat result lists and hierarchies created through traditional hierarchical clustering.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.1.1 [Models and Principles]: Systems and Information Theory

General Terms

Algorithms, Experimentation

Keywords

Interactive information retrieval, Information gain, Hierarchical clustering, Active learning

1. INTRODUCTION

The explosive growth of the number of documents accessible via online information systems has intensified the need for navigation means that allow efficient access to the documents sets. Nowadays

*This research is supported as ToKen2000 project by the Netherlands Organization for Scientific Research (NWO) under project number 634.000.006.

†This paper is an extension of the work presented on the IJ-CAI'05 workshop on Intelligent Techniques for Web Personalization (ITWP'05)

many navigation systems are publically available. Most of these use a form of keyword search or hierarchical menus. Search engines like Google [5] or AltaVista [2] and web portals like Yahoo! [25] aim at providing access to the whole web. Other systems are limited to documents on certain topics or to the contents of one web site. The common goal of these systems is to help the users reach their target information as fast as possible.

Providing access to web pages requires two steps. First, one has to determine which pages might be the user's target pages. The second step involves the creation of a suitable structure to present links to the candidate pages to the user. Research on search engines typically focuses on the first step: finding the set of pages that best matches the user's query. Recommender systems also address this step using a user profile instead of a search query. In this work we address the second step: automatically creating a hierarchy that allows users to efficiently access the candidate links.

Sets of candidate links can be structured in many different ways. Most search engines present their search results on a number of result pages containing ordered lists of links. In some cases search engines apply clustering techniques to group the search results per topic (e.g. [26, 15, 7, 27]). Deeper hierarchical structures are common in menus of single web sites and web directories. The question that arises is: which structures result in the shortest retrieval times?

The structures that are generated in this work are targeted at users with specific information needs. These users navigate through the provided information structure looking for the pages that are relevant for their purposes, their target pages. In this case the efficiency of a navigation structure is determined by the amount of browsing required to reach the target pages. In a flat list browsing consists of choosing the best link among a set of alternatives. In a hierarchy a series of categories need to be selected.

The optimal shape of a link hierarchy is not the same in all situations. The expected efficiency of a navigation structure depends on the probability that the candidate links are targets. If the system knows almost for sure that the user is interested in certain links, the best strategy is to show these links immediately. In other words, to structure the links as a flat list. On the other hand, if there are many links that have an equally small probability of being a target, a deeper hierarchy can be more efficient. Through the selection of categories in the hierarchy the user provides information about his or her target. This information is used to reduce the number of candidate links the user needs to consider.

In this work we present an algorithm that weights the time needed

to choose a category against the expected gain of providing extra information. At each step in the interaction with the user the algorithm computes the probability that each page is the user's target page. The categories and links with the highest expected information gain are presented. The resulting document hierarchy minimizes the number of clicks the user needs to make to reach his target pages. We demonstrate in a small scale experiment that users need less clicks when they use the hierarchies created by this method than when they use flat lists of links. Moreover, simulation experiments indicate that the created structures are more efficient than document hierarchies created through content clustering.

Section 2 discusses related work on optimizing document hierarchies. Section 3 describes the problem that is addressed in this work. In section 4 we discuss how the most informative sets of links and categories are selected. In section 5 we present the results of the experiments. The last section contains conclusions and discusses our results.

2. RELATED WORK

Related work can be classified into two categories. First, we give a brief overview of metrics for measuring the efficiency of hierarchical link structures. Afterwards, we discuss methods to automatically create and optimize these structures.

Many researches have studied the relation between of the structure of a hierarchy and the time that users need to retrieve items. A majority of the authors find a linear relation between retrieval time and the number of clicks necessary to reach a content page [10, 22, 14]. A linear relation is also the most common choice in models of web navigation [11, 14, 1]. The relation between retrieval time and the number of list items per hierarchy layer depends on the organization of the lists. In an ordered list users can use binary splits so that retrieval time is roughly logarithmic in the number of list items [10, 16]. In an unordered list the relation is linear [11, 22, 14].

Web search result clustering is a common method to assist users in finding relevant links among a set of retrieved web links. After a search engine has retrieved a set of documents matching a user's query, documents with similar contents are placed under a common header. Words that occur frequently in the clusters' documents are used as cluster labels. Several authors report that with result set clustering users need less time to find the relevant information (e.g. [26, 15, 7]). In [3] the documents are not clustered but classified into a predefined hierarchy. Zeng et al. [27] extract keyphrases from the documents and form clusters of pages containing the phrases. The top ranked clusters are labeled with corresponding keyphrases and presented to the user. The advantage of this method is that it yields both query specific clusters and high quality labels. These methods have in common that they all aim at optimizing the clusters' coherence and the clusters' descriptions. To our knowledge no attempts have been made to include the probability distribution over the links and optimize the clusters from an information theoretic perspective.

Other researchers focus on estimating the probabilities that pages are targets, e.g. [9]. Their methods improve the probability distributions over the pages which enables a better ordering of the links on the result pages. However, the improved probabilities are not used to create other structures than flat lists.

Several attempts have been made to select parts of a hierarchy that

are interesting for certain users e.g. [18, 6]. These systems do not optimize the structure of the hierarchy, but only hide a part of it. Hiding nodes can improve efficiency as it allows users to reach their target pages without considering uninteresting parts of the hierarchy. However, the selected parts of the hierarchies are not necessarily the most efficient structures for the remaining nodes.

Various algorithms have been developed for improving existing hierarchies. Masthoff [12] presents an algorithm that creates a hierarchy using a number of ontologies as basic hierarchies. She uses hand crafted rules to split and merge menu items with too many or too few subitems. In [22] WAP menus are adapted to the usage of individual users. Frequently used items are moved to more prominent positions in the menu. For both methods the authors show that they can improve the efficiency of the hierarchies. However, there is not guarantee that they converge to maximally efficient structures. A method for which this guarantee can be given is presented by Witten et al. [24]. They optimize the index of a digital phonebook using the entropy of the probability distribution over the names. Their algorithm does make optimal decisions, but it only applies to domains in which the names of the searched items are known and ordered as in the case of a phonebook.

McGinty and Smyth [13] use critiquing to determine the users' targets. They argue that always presenting the links with the highest probability can cause a user to get stuck in an uninteresting part of the page space. They overcome this problem by uniformly spreading the presented links over the page space when the user seems to be making no progress. At each step they either try to maximize the probability of presenting a target or aim at collecting new information. With the approach presented in the present work one does not have to make this choice. The information gain criterion automatically results in broader categories when little is known about the user and in more specific links when more information becomes available.

3. PROBLEM SETTING

The task that is addressed in this work is to find a hierarchical structure for a set of links that minimizes a user's retrieval time. Before a system can accomplish this task it needs to compute for each page the probability that the page is a target page. Furthermore, the pages must be annotated with keywords that can serve as category labels. In this section we explain how the probabilities and keywords can be acquired in various situations. In addition, we present the interface that will be used for evaluation and discuss two popular structures that will serve as baseline structures in the experiments.

The constructed hierarchies consist of candidate links and categories. The candidate links form the terminal nodes in the hierarchy. They point directly to the candidate pages and have as anchors the pages' names. Non terminal nodes are categories. A user who selects a category goes a level deeper in the hierarchy and is presented with a new set of choices. A category node is labeled with a keyword or keyphrase that describes the contents of the pages below the node. Because all categories in the hierarchy must have a label, the available keywords determine the possible categorizations. Users navigate top down through the hierarchy opening links and categories that match their information needs. The task of the system is to place at each hierarchy layer the categories and links that minimize the average retrieval time. This task is depicted graphically in figure 1.

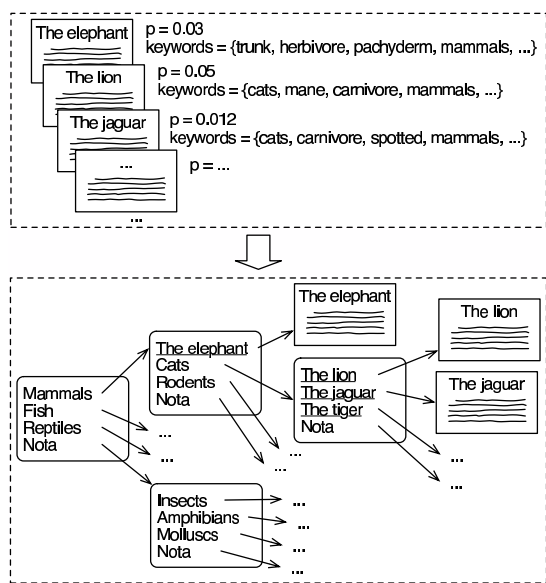


Figure 1: Example of a set of candidate pages with probabilities and keywords and a hierarchical structure for this set. ‘Nota’ is short for ‘None of the above’.

The retrieval time of a given target page depends on the location of the page in the hierarchy. In correspondence with the literature (e.g. [10, 22, 14, 11, 14, 1]) we assume that retrieval time varies linearly with the number of clicks a user needs to make before reaching her target pages (the total path length). The length of the path to a content page is equal to the depth of the page in the hierarchy since users browse top down through the hierarchy. We fix the number of links per hierarchy layer, so that we not have to make assumptions about the relation between retrieval time and the number of links per layer. Therefore, minimizing the average retrieval time reduces to minimizing the expected path length to the user’s target pages.

The structures are built for a closed set of links called the *candidate links*. Some candidate links point to the user’s target pages. The set of candidate links is available to the system, but the system does not know which links are targets. However, there is a probability assigned to each candidate link. How the probabilities are computed depends on the application. In a search engine the candidate links are the links that match the user’s query. The target pages are the pages that the user finds relevant. The probabilities can be adapted from the relevance scores of the candidate links. For a recommender system the candidate pages are the pages of the web site the system is part of. In this case, the probabilities can represent the pages’ access frequencies or the personal interests of the user.

To label the categories in the hierarchy a set of keywords is needed. The system also needs to know which keywords apply to which pages. For simplicity we assume keywords either apply or do not apply to a page, but the presented methods can be adapted straightforwardly to handle probabilistic keywords assignments. Keywords from various sources can be used to annotate the pages. If the candidate pages are already annotated with keyword meta tags, these keywords can be used directly. Otherwise, some keyword extraction mechanism needs to be deployed to extract keywords from the pages’ contents (e.g. [27]).

The interface that enables the users to browse through the hierar-

chy can have different forms. Here we assume it looks similar to the result pages of search engines like Google [5] and AltaVista [2]. This means that users do not see the whole hierarchy, but only the choices corresponding to their current position in the hierarchy. Furthermore, the number of items that is shown in each step is fixed at n . It is always possible that none of the presented links and categories applies to the user’s information needs. Therefore, there is always a choice labeled ‘None of the above’. This choice is comparable to the link that points to the next result page in a search engine. When a user follows a link to a content page, he receives the page’s contents. After reading the page he can stop the interaction or continue with a new search action.

We compare three strategies for determining which links and categories are shown at each hierarchy level. Search engines usually show flat lists of candidate links starting with the n most probable links. If the user’s target is not among the presented links she clicks ‘None of the above’ and receives the next probable links. This structure maximized at each step the probability that the user can reach her target directly. For this reason we call it the *greedy strategy*. *Clustering based strategies* form hierarchies of page clusters on the basis of similarities between the pages. The keywords are used to label the clusters and create categories. In the next section we will present a new strategy called *the information gain strategy*. This strategy selects the links and categories with the highest information gain. It does not maximize the probability of showing a target link, but the information gained in each step.

4. THE INFORMATION GAIN STRATEGY

In this section we explain how the information gain strategy selects the most informative categories and links. In section 4.1 we discuss the computation of the expected information gain of a set of categories and links. Section 4.2 covers heuristics to find sets with high information gain.

4.1 Category Information Gain

In theory the most efficient structure can be determined completely. With the page probabilities we can compute the probability that a user is looking for a page from a certain category. If we make the assumption that users select with some probability the categories that contains their goal pages, we can compute the probability that a category is selected when it is presented to the user. We can write down all possible navigation traces for all possible structures and compute the lengths and probabilities of the traces. Now we just select the structure with the shortest expected path length.

This strategy always results in the optimal path lengths, but unfortunately it is not tractable in practice. We need a more efficient category selection algorithm, especially when all computation must be done while the user is waiting for his page. Often the structure can not be built in advance, for instance because the candidate set depends on a search query or because the target probabilities are adapted at run time to the user’s interests.

To deliver the structures in reasonable time we create the structures top down only expanding nodes that are actually visited. Moreover, the navigation structures are optimized node by node instead of globally. At each step the system selects a set of links and categories without considering all possibilities for the deeper hierarchy layers. Below we explain the criterion according to which the information gain strategy selects the category sets. This criterion does not distinguish between links and categories. Links are treated as categories containing exactly one page.

To select the best category set we need to estimate the users' path lengths when a particular set of categories and links is presented. A measure which does exactly this is the *information gain* [19]. The information gain of a category tells us how much knowledge we have gained if the user selects the category. This depends both on the number of pages in the category and the candidate link probabilities. The expected information of a set of categories and links is the expected amount of information that is gained when the set is presented to the user.

The following example illustrates the working of the information gain criterion on the selection of categories for a set of pages about animals. If our only knowledge is that a user searches for an animal, broad categories like 'mammal' and 'fish' are informative. A click on one of these categories tells us in what kind of animal the user is interested. However, if for some reason we expect that the user is looking for a furry animal, the selection of 'mammal' does not provide much new information. In this case more information is gained by presenting narrower categories like 'rodent' and 'nocturnal animal'. Another important point is that showing two distinct categories like 'mammal' and 'fish' provides more information than showing largely overlapping categories like 'fish' and 'water animal'.

Formally, the information gain of a question is the difference between the number of bits of information needed to determine the target before and after asking the question. The expected information gain, IG , of a set of categories and links L is given by:

$$IG(L) = H(P) - \sum_{\{l \in L\}} (p(l|L) * H(P|l))$$

Here P is the probability distribution over the set of pages D . $p(l|L)$ is the probability that the user chooses category or link l provided that the items from L are presented. $H(P)$ gives the entropy of P . $H(P|l)$ is the entropy of the probability distribution after l has been chosen. $H(P)$ is given by:

$$H(P) = -\sum_{\{d \in D\}} (P(d) \log(P(d)))$$

The distribution $P|l$ depends on the node type of l . If l is a link, the selection of l provides certain knowledge that the user was interested in following link l . In other words, the probability of l becomes 1 and the remaining entropy, $H(P|l)$, is 0. If l is a category, the selection of l does not provide certain knowledge about the user's target, but does provide evidence that the user's target belongs to category l . One possibility is to assign zero probability to all pages that are not annotated with the selected category. However, even if the keyword annotations are chosen carefully it can happen that a user finds that a page belongs to a category that is not present the annotation. Therefore we use an update mechanism that ensures that page probabilities are adjusted according to the selected categories, but never become zero. For details on this mechanism see [8].

The a priori candidate link probabilities are used to select the items that are shown at the root of the hierarchy. To create the deeper hierarchy layers the knowledge gained from the selected categories is incorporated in the probabilities. For instance, for the selection of the nodes below the category 'mammal', we use the probability distribution $P|mammal$ as base probabilities.

The information gain strategy selects the set of categories and links with the highest information. A high information gain means that the uncertainty that is left in the probability distribution is low. This

indicates that after the users selects a category we need only a small number of steps to get perfect knowledge about the users target. Thus, selecting the category set with the highest information gain on average leads to the shortest path lengths for the user. Note, that this strategy leads to optimal categories for each step, but whether these choices are optimal overall depends on the categorizations available for the later steps.

4.2 Finding Informative Category Sets

The information gain criterion allows us to estimate how much a set of categories and links will shorten the path length without considering all possible continuations of the interaction. Unfortunately, this still does not make the problem tractable. If the number of links that can be presented on a page is n and the total number of categories and links is k , then the number of possible sets is n^k . Because this number can be prohibitively large, in this section we present heuristics to preselect some promising sets. The heuristics do not simplify the computation of the sets' information gain, but reduce the number of sets for which the information gain is computed.

As a first filter we throw out categories with a very small or very large probability of being chosen. If a category is associated with only one page it is obviously better to provide a direct link to the page than to show the category. Therefore, we compute for each category the probability that it contains a target page and throw out all categories with a probability smaller than the probability of the n th most probable page. Furthermore, if it is almost certain that the target belongs to some category, then selecting this category does not provide much new information. For this reason categories with a very large probability are also filtered out.

In a pilot study [8] we compared two heuristics for finding the best set among the categories and links that remain after filtering. The heuristic that proved most effective uses a form of hill climbing. It computes the information gain of all sets containing only one category or link. The n categories or links with the highest information gain are used as start set (n is the allowed number of links per page). One item from the start set is exchanged for another category or link. If this results in a set with a higher information gain the change is pertained; otherwise it is undone. This exchange process is repeated until no more changes can be tried or until a maximum number of steps is reached. The resulting set is presented to the user. Like all hill climbing methods this heuristic can converge to local maximum, but experiments show that in practice it finds good category sets.

5. EVALUATION

5.1 Experimental Setting

To evaluate the information gain strategy, the greedy strategy and the content clustering strategy we measure the efficiency of structures generated by each of the strategies in a series of experiments. In these experiments we use a fixed set of candidate pages and two versions of the probability distribution: a static distribution and a distribution that reflects the users' previous targets.

The candidate set is comprised of the combined sets of pages of two Dutch web sites for elderly people: the SeniorGezond site [21] and the Reumanet site [20]. Both sites were developed by The Netherlands Organization for Applied Scientific Research (TNO) in cooperation with domain specialists from the Geriatric Network and the Leiden University Medical Center. SeniorGezond contains

information about the prevention of falling accidents. Reumanet contains information about rheumatism. The sites have very similar structures: they consist of a set of short texts describing a particular problem or product and a hierarchically structured navigation menu. The menu provides information about the relations between the pages, but each text is written in such a way that it can also be understood in isolation. From all pages of the two sites we removed the navigation menu and all in text links. Fifteen texts that were in almost the same form present on both sites were mapped onto one page. After this mapping 221 unique pages remained, each consisting of a title and some flat text.

A semi-automatic method was used to assign keywords to the pages. We manually created a domain specific ontology consisting of 800 terms or phrases and a broader term - narrower term relation. The terms from the ontology were automatically assigned to the pages. We counted for each text and each term in the ontology the evidence that the term was a keyword for the text: the number of times the term or one of its descendants appeared in the text. The pages were annotated with all terms with an evidence of at least 2. The domain specific ontology was created by hand, because there was no ontology available for the domain and many of the domain specific keywords were not in the Dutch version of WordNet [4]. The average number of keywords of a page was 7.7.

The quality of the keywords was evaluated in a survey [8]. We found that on average the participants labeled 36% of the keywords in the annotation as not appropriate for the texts. Apparently the precision of the annotation procedure was not very high. Using these keywords as category labels in a navigation structure will probably cause the users to follow a considerable amount of incorrect paths. In the next section we will see how this effects the efficiency of the information gain strategy. Another interesting finding from the keyword evaluation was that there was 80% agreement between the answers of the various participants. This suggests that it is possible to learn the associations between pages and keywords from the behavior of the users. This allows a system to automatically improve the pages' annotations. We plan to explore this idea further in the future.

To decrease the influence of the chosen probability distribution we tested structures that were generated with two different distributions. The first version is a static distribution. In the server logs of the two web sites we counted the number of requests for the candidate pages. From the request frequencies we constructed the a priori probabilities. When users searched for more than one target page, for each search the same a priori probabilities were used to build the navigation structure. Thus, each search started at the root of the same hierarchy.

For the second distribution we used a form of personalization. Each time a user reached a content page the probabilities of the candidate pages were adapted. When the users decided to search further the adapted probabilities were used to build a new navigation structure. The new searches still started at the root of the hierarchies, but the hierarchies became more and more personalized. The personalization process increased the probability of pages that were similar to the visited content page and decreased the probability of dissimilar pages (for details see [8]). The similarity between two pages was computed as the minimal conditional probability of the pages [17]. Like the a priori probabilities the conditional probabilities were taken from the server logs. For the distances of pages from different sites we used a content based measure.

<p>Task: glasses and contact lenses</p> <p>You have difficulty reading and you think you might need glasses. Find as much information as possible on (buying) glasses and contact lenses.</p> <hr/> <p>Target pages: Optician.htm Seeing+and+hearing.htm</p>

Figure 2: Translated example task with target pages. The target pages were not visible to the participants.

We defined 12 search tasks for which information could be found in the candidate pages. A task consisted of a short description of a specific problem of an elderly person. The users had to search all pages related to the problem. The topics of the tasks were chosen after consultation of the creators of the sites. We tried to choose problems that were realistic in the domain to get a realistic simulation of the site's users. We defined by hand which pages were in the target sets for the tasks. The tasks had between 2 and 12 target pages. An example of a task description is shown in figure 2.

5.2 Experiments

To show the advantage of the information gain strategy over the greedy strategy and the clustering based strategy we performed a series of experiments. The three strategies were used to build hierarchies on the basis of the static and the personalized probability distributions. This resulted in five structures: a greedy structure, a personalized greedy structure, a clustering based structure, an information gain structure and a personalized information gain structure. The structures were not built in advance, but expanded each time a user opened a node.

The structures were built as described in the previous sections. The greedy structures consisted of ordered lists of pages. The most probable pages were located at the root of the hierarchy, the next most probable pages at the second level etc. For the creation of the clustering based structure we used a divisive form of k-means comparable to the bisecting k-means algorithm of [23]. This algorithm split the set of pages at each level in a number of clusters. For each cluster the best matching keyword was used as category label. Pages that did not belong to one of the resulting categories were placed under the 'None of the above' category. The similarity measure was the same one we used for personalization in section 5.1. The clustering based structure was constructed only on the basis of the page similarities and did not discriminate between the two probability distributions. The information gain hierarchies had on each node the most informative categories.

We created an interface that allowed users to browse through the hierarchy. The interface functioned as explained in section 3. The number of links or categories on a page was set at 5, including the 'None of the above' category. This number was purposely chosen to be quite low, so that many clicks would be required to reach the targets. In a perfectly balanced hierarchy 221 pages are reachable in 4 steps. In a greedy structure the 221 pages are divided over 56 result pages.

In the first part of the experiment we evaluated the efficiency of the structures with simulated users. The simulated users had a set of pages which were their target pages. They traversed the hierarchy looking for their targets. They never went to content pages which

Method	No. steps
Greedy	27.7
Personalized greedy	9.0
Clustering	15.1
Information gain	8.2
Personalized information gain	4.6

Table 1: The average number of steps of simulated users with perfect choices.

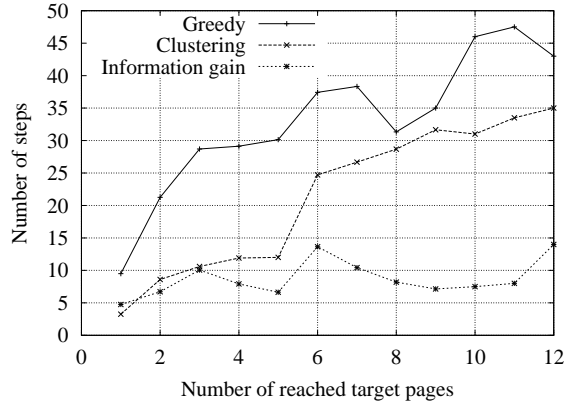


Figure 3: The average number of steps that simulated users with perfect choices needed to reach each of the target pages.

were not in their target set and when a link to a target page was available they always went there directly. When no links to target pages were available they considered the available categories. If one of the categories matched a target, they opened the category. When also no relevant categories were shown, they clicked ‘None of the above’. They kept searching until all targets were found.

We compared simulated ‘perfect’ users that always chose the correct categories with ‘imperfect’ users that sometimes followed an incorrect path. The imperfect users sometimes chose categories that were not related to their targets or ‘None of the above’ when an appropriate category was presented. We did not add noise to the content link choices, because we assumed that users could accurately judge the relevance of pages from their titles.

We evaluated the real world value of the greedy and the information gain structures in an experiment with real users. Thirteen participants were asked to perform all 12 multiple target search tasks. The participants got only the topics of the tasks and not the sets of target pages. Every participant used two of the four structures, one during the first 6 tasks and another during the next 6 tasks. The order of the tasks and the structures was varied over the participants. We measured the number of clicks the users needed to find the targets and the number of relevant pages that were found.

5.3 Results

5.3.1 Simulation

Table 1 gives the average number of clicks that the simulated users needed to reach their targets. All figures are averages over 25 runs. With both probability distributions perfect users needed a significantly¹ smaller number of steps in the information gain structure

¹In the simulation experiments significance is computed with a two tailed paired t-test with a confidence level of 0.95.

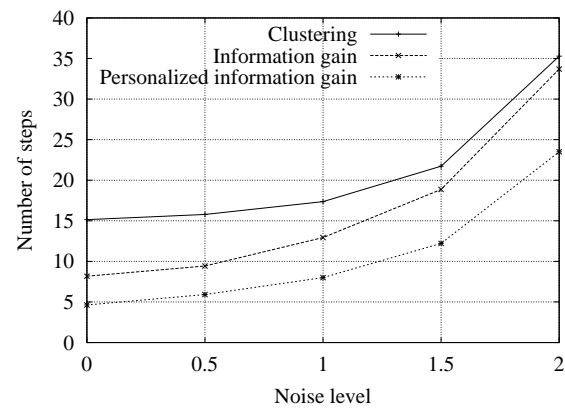


Figure 4: The average number of steps of simulated users with various noise levels.

than in the greedy structure and the clustering structure. Personalization made the greedy and the information gain strategy significantly more efficient. With personalization the difference between the two strategies is smaller, but the information gain strategy is still 49% faster. Personalization does not effect the clustering structure, because the clustering strategy does not use the page probabilities.

Figure 3 shows the number of steps that were needed to reach the various target pages. The greedy and the clustering structures provided short paths to the first targets, but when the users searched further the paths became very long. The path lengths in the information gain structure were more stable.

In the next experiments we looked at the behavior of users with various amounts of incorrect choices. The results are presented in figure 4. In this figure a noise level 1 correspond to the values found in the keyword evaluation survey (see section 5.1). The incorrect categories had a probability of 0.013 to be opened. The probability of clicking ‘None of the above’ when an appropriate category was presented was 0.36. For the other noise levels these values were multiplied by the noise level. The efficiency of the greedy structures is not shown. These structures do not use categories and therefore are insensitive to the type of noise that was added.

Figure 4 shows that the efficiency of the category structures decreased rapidly when the users made more mistakes. At the highest noise levels the path lengths become even longer than the path lengths in the greedy structures. The information gain structures performed better than the clustering structure at all noise levels. However, the influence of the amount of incorrect choices appeared to be much larger than the influence of the type of navigation structure. This suggest that information value is a useful criterion to choose between categories with equally good labels, but that the highest priority must be to given to finding categories with high quality labels.

5.3.2 Human Search

Table 2 and Figure 5 show the results of the experiments with real users. Users of the information gain structures needed significantly² less steps to reach the targets than users of the greedy structures.

²In the experiments with real users significance is computed with a two tailed t-test with a confidence level of 0.95.

Method	No. steps	No. targets
Greedy	17.8	0.9
Personalized Greedy	9.8	1.7
Information gain	11.1	1.5
Personalized information gain	6.9	1.4

Table 2: The average number of steps of human users and the average number of targets that were found by human users.

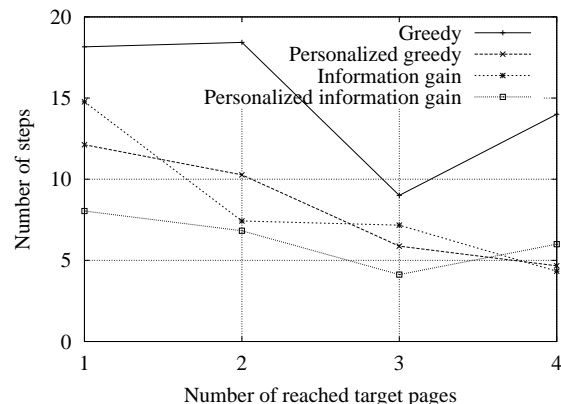


Figure 5: The average number of steps that human users needed to reach their target pages.

The relevance of the categories in the information gain structure was not always clear to the participants which led to suboptimal paths. 17% of the chosen keywords were not in the target pages’ annotations. In 7% percent of the cases in which ‘None of the above’ was clicked, there actually was a relevant category among the presented keywords. Because of these ‘mistakes’ the participants’ paths were much longer than the optimal path lengths measured in the previous section. However, the shorter path lengths of the information gain structure show that even with these high noise levels categorization can be effective.

Personalization significantly reduced the number of steps of both the greedy and the information gain structure. In contrast to what we saw in the simulation experiments Figure 5 shows that personalization not only helped during the later stages of the search, but also reduced the number of clicks needed to find the first target. This is caused by the fact that the real users sometimes clicked on links to pages that were not relevant for the task. Apparently, according to our distance function these pages were close to the target pages, so that the adaptive strategies could lead the users efficiently from these pages to the nearby targets.

We did not find large differences between the numbers of target pages that were found. The only significant result was that users found more targets when assisted with the personalized greedy structure than with the greedy structure. Probably users of the greedy structure were tempted to give up when they saw that they would have to go through the same lists of links again. With all structures the users found very few targets. Most likely this is a consequence of the limited interface. Many participants reported that they had trouble judging how many relevant pages were available, because the interface did not provide an overview of the candidate pages. This problem is less likely to occur on real sites where more information is available like the number of search result.

In conclusion, the simulation experiments showed that maximizing the information gain reduces the length of the paths to the users’ target pages provided that the category labels are sufficiently clear. The experiments with human participants show that users are able to make effective use of keyword structures and thus need less clicks in an information gain structure than in a greedy structure. In this work we used a simple method to compute the page probabilities, but the information gain criterion can be used without modification on top of more advanced page probability estimators. Our findings suggest that in any case maximizing the information gain will effectively balance the collection and exploitation of knowledge and so minimize the users’ path lengths.

6. CONCLUSION AND DISCUSSION

In a variety of domains systems create hierarchical structures for sets of candidate pages. Search engines and recommender systems typically return series of pages with flat lists of links. At each page they maximize the probability of showing a target link by presenting the most probable candidate links. They focus entirely on using their current knowledge about the user to determine which pages are the most likely targets. In other words, they follow a greedy strategy. Clustering based method do not only show links but also categories. However, these categories are not chosen to minimize the users’ path lengths, but to group the most similar pages.

In this work we present a method that actively minimizes the length of the user sessions balancing the costs of collecting more information by showing categories against the expected gain of the extra knowledge. Evaluation with artificial and experimental data shows that this information gain strategy effectively reduces the users’ numbers of clicks compared to the greedy and the clustering based strategy.

The advantage of maximizing information gain is independent of how we estimate the probabilities that pages are targets. In this work we used simple algorithms for estimating the page probabilities. More advanced methods, such as the one presented in [9], can make the estimations more accurate which leads to more efficient structures. However, these better estimations improve both the greedy and the information gain structure. To demonstrate this we tested structures generated on the basis of two probability distributions: a static and a personalized distribution. Experiments showed that the extra knowledge provided through personalization reduced the number of clicks in the greedy structure as well as the information gain structure. Improving the estimation accuracy improves efficiency but does not lessen the need for active knowledge collection.

The number candidate pages that we used for the experiment was quite small compared to the number of pages of an average web site or the average number of search results. To be able to measure the effect of categorization the number of links or categories per page was also kept low. Although the theoretical advantage categorization is independent of the size of the domain, more research is needed to show the practical value of the information gain strategy in realistic domains. We are currently incorporating the information gain and the greedy strategy in recommender systems that will be included in the real version of the SeniorGezond site. Running these systems in parallel allows us to compare the value of the strategies in a real world application.

Until now we have not considered the order in which the links are shown on the pages. Especially when the lists are long this is not

realistic, because users do not always consider all available options before clicking a link. This means that on average users need less time to choose top ranked links than to choose links at lower positions. Greedy strategies accommodate for this phenomenon by ordering the links according to their probability. The information gain strategy does not currently include link order. It can present the categories in order of probability, but how the categorization itself should be adapted is an open issue.

The candidate pages in the evaluation were annotated with terms from a manually created ontology. Many of the keywords in the annotations were ambiguous so that the participants made a considerable amount of suboptimal selections. We showed that despite these 'mistakes', the categorizations chosen by the information gain strategy shortened the users' path lengths. However, simulation experiments demonstrated that when users make too many mistakes presenting categories can reduce efficiency. These results suggest that a category's label quality is at least as important as its information value. Consequently, in a real application one should be careful only to use categories for which an adequate label is available. We are currently exploring possibilities to automatically determine the quality of category labels.

7. REFERENCES

- [1] J. Allan, A. Feng, and A. Bolivar. Flexible intrinsic evaluation of hierarchical clustering for tdt. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 263–270, New Orleans, USA, 2003.
- [2] AltaVista search engine. <http://www.altavista.nl>.
- [3] H. Chen and S. Dumais. Bringing order to the web: automatically categorizing search results. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, The Hague, The Netherlands, 2000.
- [4] EuroWordNet. <http://www.illc.uva.nl/eurowordnet/>.
- [5] Google search engine. <http://www.google.com>.
- [6] P. Haase, A. Hotho, L. Schmidt-Thieme, and S. Y. Collaborative and usage-driven evolution of personal ontologies. In *Proceedings of the 2nd European Semantic Web Conference*, pages 486–499, Heraklion, Greece, 2005.
- [7] M. Hearst and J. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, 1996.
- [8] V. Hollink, M. Van Someren, S. Ten Hagen, and B. Wielinga. Recommending informative links. In *Proceedings of the IJCAI-05 Workshop on Intelligent Techniques for Web Personalization (ITWP'05)*, Edinburgh, UK, 2005.
- [9] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, Edmonton, Canada, 2002.
- [10] T. Landauer and D. Nachbar. Selection from alphabetic and numeric menu trees using a touch screen: Depth, breadth and width. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 73–78, San Francisco, USA, 1985.
- [11] E. Lee and J. MacGregor. Minimizing user search time in menu retrieval systems. *Human Factors*, 27(2):157–162, 1985.
- [12] J. Masthoff. Automatically constructing hierarchies: An algorithm and study of human behaviour. *Forthcoming*.
- [13] L. McGinty and B. Smyth. Tweaking critiquing. In *Proceedings of the Workshop on Intelligent Techniques for Personalization as part of The Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.
- [14] G. Miller and R. Remington. Modeling information navigation: Implications for information architecture. *Human-Computer Interaction*, 19:225–271, 2004.
- [15] R. Osdin, I. Ounis, and R. White. Using hierarchical clustering and summarisation approaches for web retrieval: Glasgow at the TREC 2002 interactive track. In *Proceedings of the Eleventh Text REtrieval Conference*, Gaithersburg, USA, 2002.
- [16] K. Paap and R. Roske-Hofstrand. The optimal number of menu options per panel. *Human Factors*, 28(4):377–385, 1986.
- [17] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118:245–275, 2000.
- [18] D. Pierrakos and G. Paliouras. Exploiting probabilistic latent information for the construction of community web directories. In *Proceedings of the 10th International Conference on User Modeling*, pages 89–98, Edinburgh, UK, 2005.
- [19] J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
- [20] Reumanet. <http://www.reumanet.nl/>.
- [21] SeniorGezond. <http://www.seniorgezond.nl/>.
- [22] B. Smyth and P. Cotter. Intelligent navigation for mobile internet portals. In *Proceedings of the IJCAI'03 Workshop on AI Moves to IA: Workshop on Artificial Intelligence, Information Access, and Mobile Computing*, Acapulco, Mexico, 2003.
- [23] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proceedings of the KDD Workshop on Text Mining*, Boston, USA, 2000.
- [24] I. Witten and J. Cleary. On frequency-based menu-splitting algorithms. *International Journal of Man-Machine Studies*, 21:135–148, 1984.
- [25] Yahoo! web directories. <http://dir.yahoo.com/>.
- [26] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. In *Proceedings of the Eighth International World Wide Web Conference*, Toronto, Canada, 1999.
- [27] H. Zeng, Q. He, Z. Chen, W. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, Sheffield, UK, 2004.